

**БЕЛКООПСОЮЗ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БЕЛОРУССКИЙ ТОРГОВО-ЭКОНОМИЧЕСКИЙ
УНИВЕРСИТЕТ ПОТРЕБИТЕЛЬСКОЙ КООПЕРАЦИИ»**

Кафедра информационно-вычислительных систем

**Е. А. ЛЕВЧУК
Т. А. ЗАЯЦ**

**АВТОМАТИЗАЦИЯ
ЭКОНОМИЧЕСКИХ РАСЧЕТОВ**

**Пособие
для студентов экономических специальностей**

Гомель 2008

УДК 697.31
ББК 32.973.26-018.2
Л 34

Рецензенты: А. В. Воруев, канд. техн. наук, доцент кафедры АСОИ Гомельского государственного университета им. Ф. Скорины;
Н. В. Лацкевич, канд. экон. наук, зав. кафедрой статистики Белорусского торгово-экономического университета потребительской кооперации

Рекомендовано научно-методическим советом учреждения образования «Белорусский торгово-экономический университет потребительской кооперации». Протокол № 1 от 10 октября 2006 г.

Левчук, Е. А.

Л 34 Автоматизация экономических расчетов : пособие для студентов экономических специальностей / Е. А. Левчук, Т. А. Заяц. – Гомель : учреждение образования «Белорусский торгово-экономический университет потребительской кооперации», 2008. – 252 с.
ISBN 978-985-461-526-4

Представлен теоретический и практический материал по использованию инструментария табличного процессора Excel для решения экономических задач, анализу данных средствами табличного процессора Excel, импортированию данных для автоматизации экономических расчетов, автоматизации экономических расчетов с использованием VBA. Пособие предназначено студентам вузов, а также всем тем, кто решил самостоятельно освоить принципы управления и обработки числовой и символьной информации, представленной в табличной форме произвольной структуры.

УДК 697.31
ББК 32.973.26-018.2

ISBN 978-985-461-526-4

© Левчук Е. А., Заяц Т. А., 2008
© Учреждение образования «Белорусский торгово-экономический университет потребительской кооперации», 2008

ВВЕДЕНИЕ

Пособие может быть использовано для изучения следующих дисциплин:

- «Автоматизация экономических расчетов» специальностей «Управление информационными ресурсами», «Экономическая информатика», «Экономика и управление на предприятии».
- «Программное обеспечение обработки экономической и деловой информации» специальности «Управление информационными ресурсами».
- «Компьютерные информационные технологии» всех экономических специальностей.

Задача данных дисциплин заключается в приобретении студентами знаний о принципах управления и обработки числовой и символьной информации, представленной в табличной форме произвольной структуры. В соответствии с базовыми программами обучения по дисциплинам «Автоматизация экономических расчетов» и «Программное обеспечение обработки экономической и деловой информации» в пособии представлен теоретический и практический материал по следующим разделам:

- Использование инструментария табличного процессора Excel для решения экономических задач.
- Анализ данных средствами табличного процессора Excel.
- Импорт данных для автоматизации экономических расчетов.
- Автоматизация экономических расчетов с использованием VBA.

В настоящее пособие авторы включили множество примеров различной степени сложности. Заслуживает особого внимания стиль их изложения. Как правило, после точной и детальной постановки задачи приводится не полное решение, а только его ключевые шаги в табличном виде или средствами иллюстраций копий экрана. Студентам необходимо самостоятельно продумать решение, найти подходящий способ реализации и затем сравнить с авторскими результатами. Это стимулирует развитие творческих способностей, требует глубокого осмысления теоретических сведений, воспитывает нешаблонное мышление.

Раздел 1. ИСПОЛЬЗОВАНИЕ ИНСТРУМЕНТАРИЯ EXCEL ДЛЯ РЕШЕНИЯ ЭКОНОМИЧЕСКИХ ЗАДАЧ

1.1. Назначение и классификация электронных таблиц

По всем миру ежедневно используются миллионы электронных таблиц. Их можно разделить на две категории:

- электронные таблицы личного пользования;
- электронные таблицы для специалистов.

Электронные таблицы личного пользования создаются для того, чтобы быстро решить поставленную задачу или ответить на какой-то вопрос. Пользователя не интересует внешний вид электронной таблицы, сложность ее построения, оформление документа. В данном случае требуется получить правильный ответ на вопрос или отобразить данные графически при минимальных затратах времени. Например, оценить ежемесячные расходы при покупке телевизора в кредит.

Электронные таблицы для специалистов предназначены для продолжительного применения специалистами различного профиля. Их условно можно разделить на следующие категории:

- отчеты и презентации (проектируются для конечного результата (вывода на печать), используются возможности форматирования и построения диаграмм);
- модели финансового анализа или анализа данных (применяются для анализа бюджета и инвестиций, моделирования и статистического анализа данных; финансовый анализ результатов происходит с использованием модели «что – если», где вычисления по формулам производятся с использованием предположений);
- управление списками (используются для управления табличными базами данных в виде списков);
- доступ к базам данных (предназначены для работы с данными, которые хранятся во внешних базах данных).

1.2. Критерии качества разработки электронной таблицы

Любая электронная таблица создается для получения конечного результата и должна удовлетворять следующим критериям:

- применение для конкретных целей;
- получение правильных результатов;
- выполнение задачи конечным пользователем за минимальное время;
- возможность для пользователя на каждом шаге знать, что ему делать дальше;
- невозможность удаления или изменения пользователем важных компонентов;
- оформление формул и макросов таким образом, чтобы при необходимости их легко можно было модифицировать.

1.3. Понятие функции

Основным достоинством Excel является наличие мощного аппарата формул и функций. В состав Excel входит около 300 функций. Если этого недостаточно для решения задач, то можно добавить дополнительные специализированные функции, разработанные независимыми производителями, или создать собственные функции, используя VBA.

Функция – это вычислительная процедура, которая выполняет определенные операции над заданным значением (или значениями) и возвращает результат их обработки. Функции предназначены для объединения множества вычислений в рабочих книгах. Они используются для выполнения стандартных вычислений в рабочих книгах. Функции в Excel не только облегчают ввод данных, но и выполняют специальные расчеты. Они могут использовать координаты диапазонов, именованные диапазоны и обычные числовые значения.

Имена и аргументы функции

Каждая функция состоит из *имени функции* и *аргумента*.

Имя функции (например, СУММ, СРЗНАЧ, МИН, МАКС) указывает на ее назначение.

Аргумент (например, B2:B12) сообщает Excel, какие адреса ячеек задействованы в данной функции.

Таким образом, значения, которые используются для вычисления функций, называются *аргументами*. Значения, возвращаемые функциями в качестве ответа, – *результатами*.

В зависимости от типа используемых аргументов применяются функции:

- без аргументов;
- с одним аргументом;
- с фиксированным числом аргументов;
- с неопределенным числом аргументов;
- с необязательными аргументами.

Примером функции, не имеющей аргумента, является функция СЛЧИС(). Функция СЛЧИС() возвращает равномерно распределенное случайное число, которое больше или равно 0 и меньше 1. Термин *равномерно распределенные* означает, что вероятность появления всех получаемых с помощью этой функции случайных чисел одинакова. Эта функция возвращает новый результат каждый раз при пересчете рабочей таблицы. Однако, даже если функция не использует аргумент, круглые скобки необходимы.

Пример

Необходимо вернуть случайное число между 0 и 12.

Решение

Так как число 12 не включается в интервал возможных значений случайных чисел, то функция =СЛЧИС() * 12 преобразует интервал от 0 до 1 в интервал от 0 до 12.

В качестве аргументов функции могут использоваться ссылки на ячейку или диапазон, постоянные значения, аргументы-выражения, аргументы-функции.

Ссылки на ячейку или диапазон

При вычислениях Excel использует текущее содержание ячейки или диапазона.

Пример 1

=СУММ(A1:D20).

Пример 2

Если определено *имя диапазона* (A1:D20 имеет имя *Продажи*), то имя может использоваться вместо ссылки =СУММ(*Продажи*).

Пример 3

В некоторых случаях можно использовать в качестве аргумента функции весь столбец или строку.

С помощью функции =СУММ(B:B) суммируются все значения, находящиеся в столбце B.

Постоянные значения или аргументы-литералы

Литералом называют число или строку текста, которые непосредственно заданы в качестве аргументов функции.

Пример 1

=КОРЕНЬ(225). В качестве литерала используется число 225.

Пример 2

=ЛЕВСИМВ(A1;1). Функция имеет 2 аргумента. Возвращает строку символов, которая определяется первым аргументом (A1), длина которой определяется вторым аргументом (1). Допустим, если в ячейке A1 находится текст *Бюджет*, то функция выделит первую букву этого слова – *Б*.

Аргументы-выражения

Выражения можно рассматривать как формулы внутри функции. Excel вначале вычисляет выражение-формулу, а затем использует полученный результат в качестве значения аргумента.

Пример

КОРЕНЬ((A1^2)+(A2^2)). Вначале вычисляются выражения в скобках, а затем из него извлекается квадратный корень.

Аргументы-функции

Выражения, используемые в качестве аргумента, могут включать не только формулы, но и другие функции. Функции, которые используются в качестве аргументов других функций, называют *вложенными*. Excel сначала вычисляет значение вложенной функции, а затем значение основной функции. В формулах можно использовать до 7 уровней вложенности функций.

Пример 1

=КОРЕНЬ(СУММ(A1:D20)). Вначале вычисляется сумма диапазона A1:D20, затем из полученного значения вычисляется квадратный корень.

Пример 2

SIN(РАДИАНЫ(B9)). Функция РАДИАНЫ преобразует значение аргумента B9, заданное в градусах, в радианы, так как во всех тригонометрических функциях Excel аргументы задаются в радианах. Затем функция SIN вычисляет синус угла.

К аргументам можно применять следующие *операторы*:

- арифметические: сложение (+), вычитание (–), умножение (*), деление (/), возведение в степень (^), вычисление процента (%);
- сравнения: равно (=), не равно (<>), меньше (<), больше (>), больше или равно (>=), меньше или равно (<=);
- связи: диапазон (:), объединение диапазонов (;);
- текстового соединения: сцепление (&).

1.4. Ввод функции

В ячейку рабочей таблицы функцию можно вводить с клавиатуры (вручную) или с помощью диалоговой процедуры, называемой *Мастер функций*.

Для того, чтобы использовать функцию, нужно ввести ее как часть формулы в ячейку рабочего листа. При ручном вводе требуется соблюдение правил *синтаксиса* функции, т. е. последовательности расположения в ячейке составных частей функции (имени и аргументов). Если пользователь нарушает правила синтаксиса, Excel выдает сообщение о том, что в формуле имеется ошибка.

Правила синтаксиса при записи функций

Если функция появляется в самом начале формулы, ей должен предшествовать знак равенства, как и во всякой другой формуле.

Аргументы функции записываются в круглых скобках сразу за именем функции и отделяются друг от друга символом «точка с запятой» (;). Скобки позволяют программе определить, где начинается и где заканчивается список аргументов. Внутри скобок должны располагаться аргументы. При записи функции должны присутствовать открывающаяся и закрывающаяся скобки, при этом не следует вставлять пробелы между именем функции и скобками.

В качестве аргумента можно использовать числа, текст, логические значения, массивы, значения ошибок или ссылки. Аргументы могут быть как константами, так и формулами.

Второй способ ввода функции автоматизирует контроль синтаксиса функции. При использовании *Мастера функций* необходимо выполнить следующие действия:

1. Поместить курсор в ячейку, в которую необходимо ввести функцию. Если функция должна стать частью формулы, то установить курсор в место вставки функции в формуле.

2. На стандартной панели инструментов щелкнуть по кнопке *Мастера функций* или выполнить команду *Вставка* → *Функция*.

3. В появившемся диалоговом окне, представленном на рис. 1.1, в списке *Категория* выбрать нужную категорию функции (математические, статистические, логические и т. д.).

4. В списке *Функция* выбрать функцию, которую нужно ввести в формулу и нажать *Далее*.

5. В зависимости от типа выбранной функции появится следующее диалоговое окно, пример которого приведен на рис. 1.2, где для аргументов функции вводятся нужные значения. Названия обязательных аргументов выделены полужирным начертанием. Поле, находящееся справа от области редактирования каждого аргумента, показывает его текущее значение. После ввода всех аргументов, под ними появляется результат вычисления функции. В нижней части диалогового окна приводится краткое описание назначения функции и активного аргумента.

6. Завершает ввод функции нажатие на кнопку *Готово* в диалоговом окне.

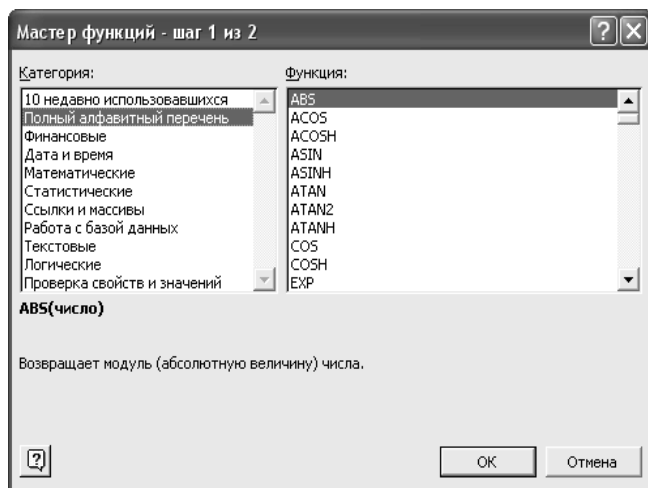


Рис. 1.1. Первое диалоговое окно Мастера функций



Рис. 1.2. Диалоговое окно функции СУММЕСЛИ

1.5. Встроенные функции

1.5.1. Категории встроенных функций

Все функции подразделяют на следующие *категории*:

- математические (СУММ, СУММЕСЛИ, СУММПРОИЗВ, ОКРУГЛ, ABS, COS, SIN, TAN, EXP, LOG и др.);
- логические (ЕСЛИ, И, ИЛИ и др.);
- ссылки и массивы (ВПР, ГПР, ПОИСКПОЗ, ПРОСМОТР, АДРЕС и др.);
- финансовые (НОРМА, ДДОБ, ДОХОД, БЗ и др.);
- информационные (ЕЛОГИЧ, ЕЧИСЛО, ЕОШИБКА и др.);
- инженерные (ДЕЛЬТА, ДФОШ, КОМПЛЕКСН и др.);
- работы с базами данных (ДМИН, ДМАКС, БДСУММ и др.);
- статистические (МИН, МАКС, СРЗНАЧ и др.);
- текстовые (ЗНАЧЕН, НАЙТИ, ПРОПИСН, СЦЕПИТЬ и др.);
- дата и время (ВРЕМЯ, ГОД, ДАТА, ДЕНЬ и др.).

1.5.2. Математические функции

Функция СУММЕСЛИ

Функция СУММЕСЛИ суммирует ячейки, удовлетворяющие заданному критерию. Диалоговое окно функции представлено на рис. 1.2.

Синтаксис: СУММЕСЛИ(Диапазон; Критерий; Диапазон_суммирования).

Диапазон – ячейки, содержимое которых должно быть отобрано по определенному критерию.

Критерий – заданный критерий отбора (текстовое значение).

Диапазон_суммирования – это фактические ячейки для суммирования. Ячейки в диапазоне суммирования суммируются в том случае, если соответствующие им ячейки в аргументе *Диапазон* удовлетворяют критерию.

Пример

Пусть ячейки A2:A5 содержат величины стоимости для четырех домов. Ячейки B2:B5 содержат величины комиссионных при продаже соответствующих домов. Нужно вычислить сумму комиссионных от продажи домов со стоимостью, большей той, что находится в ячейке A6.

Решение

Ввести в ячейку B6 формулу =СУММЕСЛИ(A2:A5;">"&A6;B2:B5) (рис. 1.3).

	А	В
1	Стоимость дома	Комиссионные от продажи
2	10 000	700
3	20 000	1 500
4	15 000	1 200
5	25 000	2 100
6	16 000	3 600

Рис. 1.3. Расчетная таблица примера функции СУММЕСЛИ

Функция СУММПРОИЗВ

Функция СУММПРОИЗВ возвращает сумму произведений соответствующих элементов массива. Диалоговое окно функции представлено на рис. 1.4.

Рис. 1.4. Диалоговое окно функции СУММПРОИЗВ

Синтаксис: СУММПРОИЗВ(Массив1; Массив2;...), где Массив1; Массив2; ... от 2 до 30 массивов, чьи компоненты нужно сначала перемножить, а затем сложить. Все массивы должны иметь одну и ту же размерность.

Пример

Пусть ячейки A2:A5 содержат значения окладов, а ячейки B2:B5 содержат количество ставок с указанным окладом. Требуется найти в ячейке B6 суммарное значение оклада с помощью функции СУММПРОИЗВ.

Решение

Ввести в ячейку B6 формулу =СУММПРОИЗВ(A2:A5; B2:B5). (рис. 1.5).

	А	В
1	Оклад	Количество ставок
2	250	1
3	150	4
4	200	3
5	125	2
6	Итого	1 700

Рис. 1.5. Расчетная таблица примера функции СУММПРОИЗВ

Функция ЦЕЛОЕ

Функция ЦЕЛОЕ возвращает целую часть числа путем отбрасывания всех дробных цифр после запятой. Диалоговое окно функции представлено на рис. 1.6.

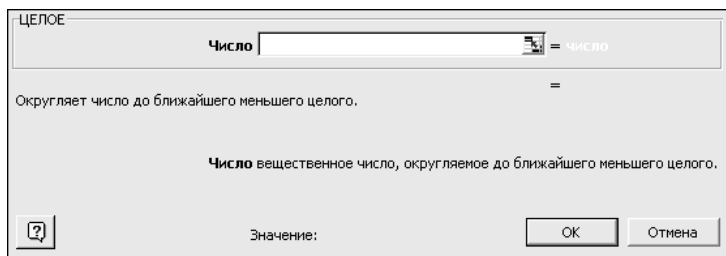


Рис. 1.6. Диалоговое окно функции ЦЕЛОЕ

Синтаксис: ЦЕЛОЕ(Число).

Пример

В ячейке A3 находится число 412,98. Требуется получить в ячейке B3 целую часть этого числа.

Решение

Ввести в ячейку B3 формулу =ЦЕЛОЕ(A3). Результат – число 412.

Функция ОКРУГЛ

Функция ОКРУГЛ округляет значение числа до указанного количества десятичных разрядов. Эта функция обычно используется для контроля точности вычислений. Диалоговое окно функции представлено на рис. 1.7.

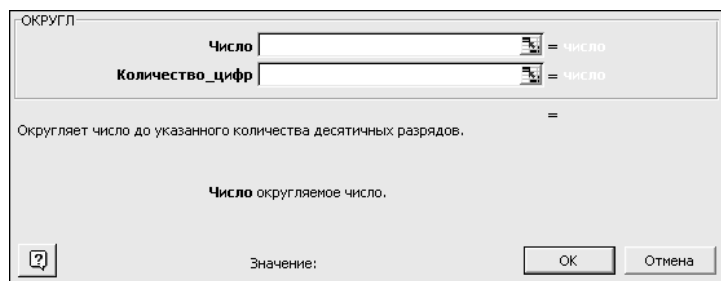


Рис. 1.7. Диалоговое окно функции ОКРУГЛ

Синтаксис: ОКРУГЛ(Число, Количество_цифр).

Число – значение для округления.

Количество_цифр – количество округляемых разрядов.

Если второй аргумент отрицательный, округление смещается влево от запятой на заданное количество разрядов.

Примеры использования функции ОКРУГЛ приведены в табл. 1.1.

Таблица 1.1. Примеры использования функции ОКРУГЛ

Функция	Результат	Функция	Результат
=ОКРУГЛ (123,457;2)	123,46	= ОКРУГЛ (123,457;-1)	120,00
=ОКРУГЛ (123,457;1)	123,50	= ОКРУГЛ (123,457;-2)	100,00
=ОКРУГЛ (123,457;0)	123,00	= ОКРУГЛ (123,457;-3)	0,00

Необходимо уметь различать округленное значение от отформатированного. Когда в формуле есть ссылка на ячейку, значение в которой было округлено с помощью функции ОКРУГЛ, будет использоваться округленное значение. Если же число было отформатировано так, что оно только кажется округленным, то в формулах со ссылками на эту ячейку будет использоваться реальное неокругленное значение.

Функции ОКРУГЛВВЕРХ, ОКРУГЛВНИЗ

Эти функции позволяют округлить значение до соответствующего кратного. Функция ОКРУГЛВВЕРХ округляет число по модулю до ближайшего большего целого, а функция ОКРУГЛВНИЗ округляет число до ближайшего меньшего по модулю целого. Диалоговое окно данных функций аналогично диалоговому окну функции ОКРУГЛ, представленному на рис. 1.7.

Синтаксис: ОКРУГЛВВЕРХ(Число, Количество_цифр); ОКРУГЛВНИЗ(Число, Количество_цифр).

Примеры использования функции ОКРУГЛВВЕРХ и ОКРУГЛВНИЗ приведены в табл. 1.2.

Таблица 1.2. Примеры использования функции ОКРУГЛВВЕРХ и ОКРУГЛВНИЗ

Функция	Результат
=ОКРУГЛВВЕРХ (123,877;2)	123,88
= ОКРУГЛВВЕРХ (123,872;2)	123,88
= ОКРУГЛВВЕРХ (123,812;1)	123,90
= ОКРУГЛВВЕРХ (123,812;-1)	130,00
= ОКРУГЛВНИЗ (123,877;2)	123,87
= ОКРУГЛВНИЗ (123,872;2)	123,87
= ОКРУГЛВНИЗ (123,892;1)	123,80
= ОКРУГЛВНИЗ (127,812;-1)	120,00

Функция РИМСКОЕ

Функция РИМСКОЕ преобразует число в его эквивалент в римских цифрах как текст. Обратное преобразование невозможно. Диалоговое окно функции представлено на рис. 1.8.

Синтаксис: РИМСКОЕ(Число; Форма).

Число – конвертируемое число в арабской записи.

Форма – число, указывающее желаемый тип числа в римской записи.

Например, функция =РИМСКОЕ(1996) возвращает результат MCMXCVI.

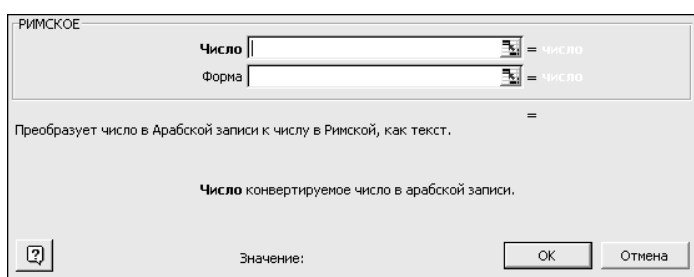


Рис. 1.8. Диалоговое окно функции РИМСКОЕ

1.5.3. Логические функции

Функция ЕСЛИ

Функция ЕСЛИ используется для проверки значений логических выражений и возврата значения в зависимости от результата проверки. Диалоговое окно функции представлено на рис. 1.9.

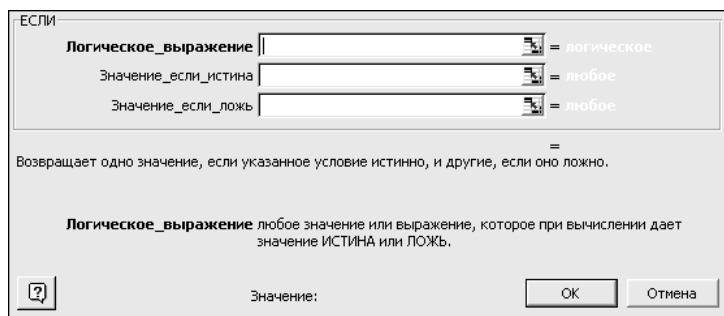


Рис. 1.9. Диалоговое окно функции ЕСЛИ

Синтаксис: ЕСЛИ(Логическое_выражение; Значение_если_истина; Значение_если_ложь).

Логическое_выражение – это любое значение или выражение, которое при вычислении дает значение *истина* или *ложь*. Вместо логического можно использовать и арифметические выражения. Нулевое значение арифметического выражения приравнивается к значению *ложь*, а ненулевое – к *истина*.

Значение_если_истина – это возвращаемый результат, если логическое выражение истинно.

Значение_если_ложь – это возвращаемый результат, если логическое выражение ложно.

Пример

Пусть ячейка B1 содержит расходы от торговли, а ячейка B2 – доходы. Требуется показать, получена прибыль (ячейка B3) или убыток (ячейка B4).

Решение

Ввести в ячейку B3 формулу =ЕСЛИ(B2>=B1;B2-B1;0), а в ячейку B4 ввести формулу =ЕСЛИ(B1>B2; B1-B2; 0).

Функция И

Функция И возвращает логическое значение *истина* или *ложь* в зависимости от логического значения своих аргументов. Если все аргументы принимают значение *истина*, то функция И возвращает значение *истина*. Если хотя бы один из аргументов функции принимает ложное значение, тогда возвращается значение *ложь*. Функция И, как правило, используется в качестве первого аргумента функции ЕСЛИ. Диалоговое окно функции представлено на рис. 1.10.

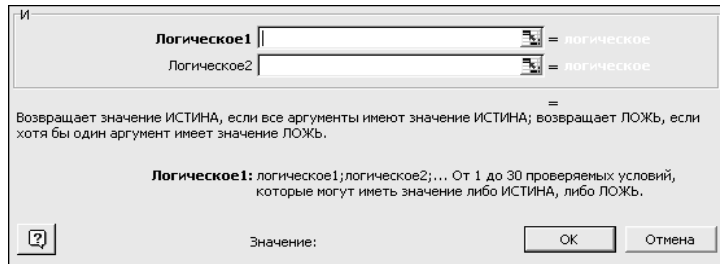


Рис. 1.10. Диалоговое окно функции И

Синтаксис: И(Логическое1; Логическое2; ...), где Логическое1; Логическое2; ... – от 1 до 30 проверяемых условий, которые могут иметь значение либо *истина*, либо *ложь*.

Пример 1

Функция =И(A1<0;A2<0;A3<0) возвращает значение *истина*, если все ячейки диапазона A1:A3 содержат отрицательные числа.

Пример 2

Коммерсант получает комиссионные в зависимости от объема продаж по обычной или премиальной ставке. Комиссионные рассчитываются по следующему критерию:

ЕСЛИ Объем продаж >= Нормированный объем продаж

И Объем расходов < Нормированный объем расходов

ТО Объем продаж * Премиальная ставка;

ИНАЧЕ Объем продаж * Обычная ставка.

Решение

В ячейке D4 находится формула: =ЕСЛИ(И(B4>=\$A\$2; C4<\$B\$2); B4*\$D\$2; B4*\$C\$2) (рис. 1.11).

	А	В	С	Д
1	Нормированный объем продаж	Нормированный объем расходов	Обычная ставка	Премиальная ставка
2	1000	300	15%	25%
3	Дата	Объем продаж	Объем расходов	Комиссионные
4	1.02.06	1 200	350	180
5	2.02.06	1 000	295	250

Рис. 1.11. Расчетная таблица примера о комиссионных

Функция ИЛИ

Функция ИЛИ возвращает значение *истина*, если хотя бы один из ее аргументов истинный. В противном случае эта функция возвращает значение *ложь*. Диалоговое окно функции представлено на рис. 1.12.

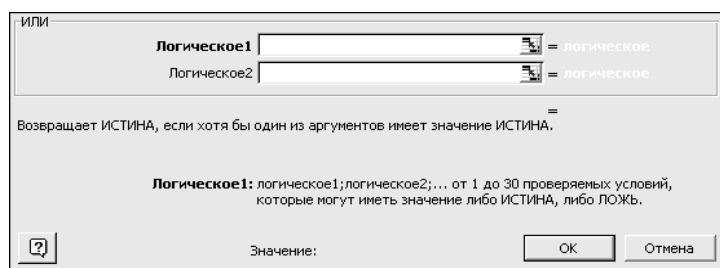


Рис. 1.12. Диалоговое окно функции ИЛИ

Синтаксис: ИЛИ(Логическое1; Логическое2; ...), где Логическое1; Логическое2; ... – от 1 до 30 проверяемых условий, которые могут иметь значение либо *истина*, либо *ложь*.

Пример

Функция =ИЛИ(A1<0;A2<0;A3<0) возвращает истинное значение, если хотя бы одно из значений, находящееся в ячейках A1, A2 или A3, меньше нуля.

Решение примера о комиссионных коммерсанта может быть получено другим способом:

ЕСЛИ Объем продаж < Нормированный объем продаж

ИЛИ Объем расходов >= Нормированный объем расходов

ТО Объем продаж * Обычная ставка;

ИНАЧЕ Объем продаж * Премияльная ставка.

Решение

В ячейке D4 находится формула: =ЕСЛИ(ИЛИ(B4<\$A\$2; C4>=\$B\$2); B4*\$C\$2; B4*\$D\$2) (см. рис. 1.11).

1.5.4. Ссылки и массивы

Функция ВПР

Функция ВПР ищет значение в первом столбце массива и возвращает значение из ячейки в найденной строке и указанном столбце. По умолчанию таблица должна быть отсортирована по возрастанию. Диалоговое окно функции представлено на рис. 1.13.

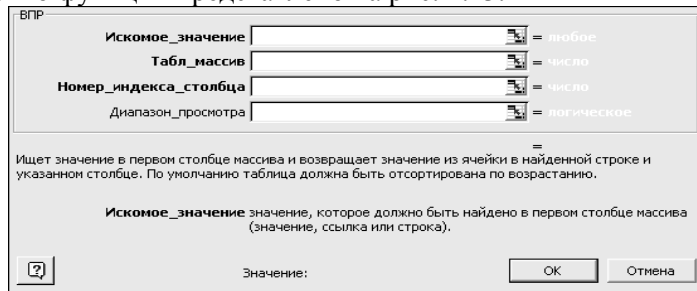


Рис. 1.13. Диалоговое окно функции ВПР

Синтаксис: ВПР(Искомое_значение; Табл_массив; Номер_индекса_столбца; Диапазон_просмотра).

Искомое_значение – это значение, которое должно быть найдено в первом столбце массива. *Искомое_значение* может быть значением, ссылкой или текстовой строкой.

Табл_массив – это таблица с информацией, в которой ищутся данные. Значения в первом столбце аргумента *Табл_массив* могут быть текстом, числами или логическими значениями. Регистр не учитывается (т. е. строчные и заглавные буквы не различаются).

Номер_индекса_столбца – это номер столбца в массиве *Табл_массив*, где должно быть найдено соответствующее значение.

Диапазон_просмотра – это логическое значение, которое определяет, нужно ли, чтобы ВПР искала точное или приближенное соответствие. Если этот аргумент имеет значение *истина* или отсутствует, то возвращается приблизительно соответствующее значение, т. е. если точное соответствие не найдено, то возвращается наибольшее значение, которое меньше, чем *Искомое_значение*. Если этот аргумент имеет значение *ложь*, то функция ВПР ищет точное соответствие. Если таковое не найдено, то возвращается значение ошибки #Н/Д.

Пример

Пусть задана таблица, приведенная на рис. 1.14.

Решение

ВПР(100; A1:C5;1;ИСТИНА) равняется 94.

ВПР(100; A1:C5;2) равняется 17.

ВПР(100; A1:C5;3;ИСТИНА) равняется 200.

ВПР(74; A1:C5;3;ЛОЖЬ) равняется 100.

ВПР(65; A1:C5;3;ЛОЖЬ) равняется #Н/Д, поскольку в первом столбце нет такого значения.

ВПР(10; A1:C5;2;ИСТИНА) равняется #Н/Д, поскольку 10 меньше, чем наименьшее значение в первом столбце.

ВПР(200; A1:C5;2;ИСТИНА) равняется 71.

	А	В	С
1	45	55	500
2	74	57	100
3	94	17	200
4	109	95	50
5	129	71	0

Рис. 1.14. Расчетная таблица для примера функции ВПР

Функция ГПР

Функция ГПР просматривает верхнюю строку таблицы и возвращает значения из указанной ячейки. Диалоговое окно функции представлено на рис. 1.15.

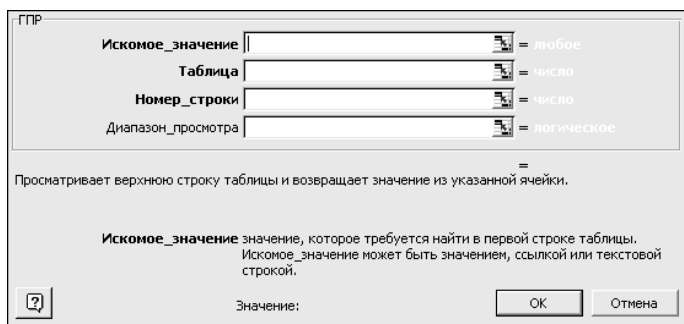


Рис. 1.15. Диалоговое окно функции ГПР

Синтаксис: ГПР(Искомое_значение; Таблица; Номер_строки; Диапазон_просмотра).

Искомое_значение – значение, которое требуется найти в первой строке таблицы. **Искомое_значение** может быть значением, ссылкой или текстовой строкой.

Таблица – таблица с текстом, числами или логическими значениями, в которой производится поиск данных, может быть также ссылкой или именем диапазона.

Номер_строки – номер строки в таблице, из которой должно быть возвращено сопоставляемое значение. Первая строка таблицы имеет номер 1.

Диапазон_просмотра – признак, который определяет, точно или приближенно должно производиться сопоставление.

Функция ГПР работает аналогично функции ВПР, с той лишь разницей, что просматривает значения ячеек первой строки таблицы (поиск по горизонтали).

Функция ПОИСКПОЗ

Функция ПОИСКПОЗ производит поиск строки текста или числа в заданном диапазоне ячеек и возвращает относительный номер строки или столбца, в котором находится найденная величина. Диалоговое окно функции представлено на рис. 1.16.

Синтаксис: ПОИСКПОЗ(Искомое_значение; Искомый_массив; Тип_сопоставления).

Искомое_значение – значение, используемое при поиске искомого значения в массиве, числе, логическом значении или ссылке.

Искомый_массив – непрерывный диапазон, возможно, содержащий искомые значения.

Тип_соответствия – число в интервале от -1, 0 или 1. Причем 0 означает точное соответствие, 1 или -1 используются для указания неточного соответствия.

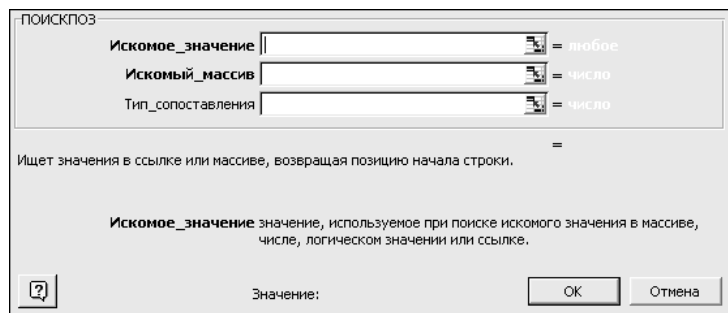


Рис. 1.16. Диалоговое окно функции ПОИСКПОЗ

Пример использования функции ПОИСКПОЗ приведен на рис. 1.17.

	А	В	С	Д
1	Искомый массив	Искомое значение	Результат	Формула
2	Январь	Июль	7	=ПОИСКПОЗ(В2;А2:А13;0)
3	Февраль	Март	3	=ПОИСКПОЗ(В3;А2:А13;0)
4	Март	Март	8	=ПОИСКПОЗ(В3;А2:А13;1)
5	Апрель	Март	3	=ПОИСКПОЗ(В3;А2:А13;-1)
6	Май	Сентябрь	9	=ПОИСКПОЗ(В6;А2:А13;0)
7	Июнь	Сентябрь	9	=ПОИСКПОЗ(В6;А2:А13;1)
8	Июль	Сентябрь	2	=ПОИСКПОЗ(В6;А2:А13;-1)
9	Август			
10	Сентябрь			
11	Октябрь			
12	Ноябрь			
13	Декабрь			

Рис. 1.17. Расчетная таблица примера функции ПОИСКПОЗ

1.6. Типы ошибок в формулах Excel

Если пользователь нарушает правила синтаксиса, Excel выдает сообщение о том, что в формуле имеется ошибка. Иногда при вводе формулы Excel выдает значение, которое начинается с символа «решетка» (#). Это говорит о том, что данная формула возвращает ошибочное значение. В подобном случае следует исправить формулу или ссылку на ячейку, содержащуюся в этой формуле, чтобы избавиться от сообщения об ошибке. Если ячейка полностью заполнена знаками решетки (#), это означает, что столбец недостаточно широк, чтобы отобразить некоторую величину. В этом случае нужно или расширить столбец, или изменить числовой формат ячейки. Типы ошибок приведены в табл. 1.3.

Таблица 1.3. Типы ошибок

Ошибочное значение	Причина ошибки
#ДЕЛ / 0!	1. Формула пытается выполнить деление на 0. 2. Формула пытается выполнить деление на содержимое пустой ячейки
#ИМЯ?	В формуле используется имя, которое Excel не может распознать (например, удалено имя, используемое в формуле, или при вводе текста где-то пропущена кавычка)
#Н/Д	Неопределенные данные. Формула ссылается на ячейку, содержащую функцию, которая ссылается на недопустимый тип данных (например, ищет минимальное значение в ячейках с текстовым типом данных)
#ЧИСЛО!	Проблема связана со значением (например, задано отрицательное число там, где необходимо положительное)
#ССЫЛКА!	Недопустимая ссылка (например, формула ссылается на ячейку, удаленную из рабочей таблицы)
#ЗНАЧ!	В формулу включен аргумент недопустимого типа

Однако, формулы могут возвращать ошибки и в том случае, если ячейки, на которые в них есть ссылки, содержат ошибочные значения. Этот эффект носит название цепной реакции, когда единственное ошибочное значение может породить ошибки во многих других ячейках, содержащих формулы со ссылкой на эту ячейку.

1.7. Связывание и консолидация рабочих таблиц

1.7.1. Понятие и назначение связывания

Связывание (linking) – это процесс использования ссылок на ячейки из внешних рабочих книг в целях получения данных для своей рабочей таблицы. Термин *связывание* служит для обозначения процесса создания формул, в которых используются значения из другой рабочей книги. Рабочие таблицы связываются вместе таким образом, что одна становится зависимой от другой.

Рабочая книга, содержащая связанные формулы (или формулы с внешними ссылками), называется *зависимой*. Рабочая книга, в которой находится информация, используемая в формуле с внешними ссылками,

ми, называется *исходной*. Причем исходная рабочая книга необязательно должна быть открыта в то время, когда открыта зависимая рабочая книга.

Можно также создавать связи с данными из других программ (например, из системы управления базами данных Access, из текстового процессора Word). Однако, в этом случае используются технологии обмена данными между приложениями – буфер обмена, динамический обмен данными, обмен связывания и внедрения. Процедура связывания особенно актуальна, если данные в исходной рабочей книге постоянно обновляются, а также в случае необходимости консолидации различных файлов.

Например, каждый региональный менеджер по продажам хранит данные в отдельной рабочей книге. Можно создать итоговую рабочую книгу продаж всех регионов, в которой будут использоваться связанные формулы в целях получения необходимых данных из каждой рабочей книги отдельного региона и вычисления итогов по всем регионам.

1.7.2. Создание формул с внешними ссылками

Существует несколько способов создания формулы с внешними ссылками:

1. Ввод ссылки на ячейку вручную. Преимущество метода в том, что исходная рабочая книга необязательно должна быть открыта.

2. Указание ссылки на ячейку. Если исходная рабочая книга открыта, то при создании формулы с внешними ссылками можно воспользоваться стандартным методом указания.

3. Использование команды *Правка* → *Специальная вставка* → *Вставить связь*. При этом исходная рабочая книга должна быть открыта.

Общий синтаксис внешней ссылки выглядит следующим образом:
[Имя_Рабочей_книги]Имя_Листа!Адрес_Ячейки.

Если рабочая книга закрыта и не находится в текущей папке, то необходимо добавить к ссылке полный путь:

=D:\STUD\И-31\[Автоматизация.xls]Лист1!A5.

Если имя рабочей книги или имя листа в ссылке содержит один или более пробелов, то такое имя необходимо заключить в одинарные кавычки:

=D:\STUD\И-31\[Автоматизация экономических расчетов.xls]Лист1!A5.

1.7.3. Создание формулы со связью путем указания

Для создания формулы со связью путем указания необходимо выполнить следующие действия:

1. Открыть исходную рабочую книгу.
2. Активизировать ту ячейку в зависимой рабочей книге, в которой будет находиться формула.
3. Ввести формулу до внешней ссылки.
4. Активизировать исходную рабочую книгу и выбрать в ней соответствующую ячейку или диапазон.
5. Закончить ввод формулы и нажать клавишу *Enter*.

Ссылка на ячейку всегда будет абсолютной (\$A\$5). Если в дальнейшем данная формула должна быть скопирована для создания других связанных формул, то абсолютная ссылка должна быть преобразована в относительную (A5).

Если исходная рабочая книга открыта, то внешняя ссылка не содержит путь к этой книге. Однако, если книгу закрыть, то к внешней ссылке автоматически добавится полный путь к книге. При выполнении команды *Файл* → *Сохранить как* для сохранения исходной рабочей книги под другим именем, Excel автоматически заменит во внешней ссылке старое имя файла на новое.

1.7.4. Создание формулы с помощью команды «Специальная вставка»

Для создания формулы с помощью команды *Специальная вставка* необходимо выполнить следующие действия:

1. Открыть исходную книгу.
2. Выбрать ячейку (или диапазон), с которой необходимо установить связь, и скопировать ее (или его) в буфер обмена.
3. Активизировать зависимую рабочую книгу и выбрать ячейку, в которой должна находиться связанная формула. Если вставляется диапазон, то выбрать только левую верхнюю ячейку вставляемого диапазона.
4. Выполнить команду *Правка* → *Специальная вставка*, а затем нажать кнопку *Вставить связь* (рис. 1.18).

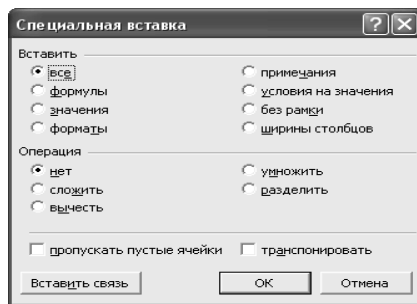


Рис. 1.18. Диалоговое окно *Специальная вставка*

Если копируется только одна ячейка, то Excel создаст формулу со связью также, как и при использовании метода указания. Однако, если копируется диапазон, то Excel создаст формулу массива, в которой будет содержаться формула со связью. *Формула массива* – это одна формула, которая хранится в нескольких ячейках. В строке формул она появится в фигурных скобках.

1.7.5. Исследование связей

Если рабочая книга связана с несколькими рабочими книгами, то может возникнуть необходимость просмотреть список всех исходных рабочих книг. Для этого используют команду *Правка* → *Связи*. В результате появляется диалоговое окно *Связи*, показанное на рис. 1.19, в котором перечислены все исходные рабочие книги, а также другие типы связей с другими документами.

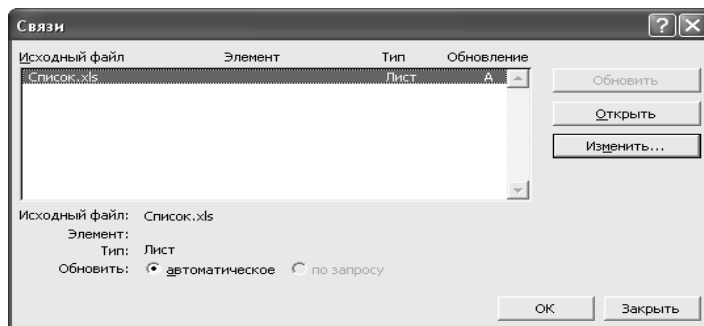


Рис. 1.19. Диалоговое окно *Связи*

Это диалоговое окно используется и для обновления связей. Для этого в нем выбирают соответствующую исходную рабочую книгу и команду *Обновить*.

В случае замены исходной рабочей книги также используется диалоговое окно *Связи*. Для этого выбирается сама исходная книга и команда *Изменить*. Появится диалоговое окно, в котором можно выбрать новый исходный файл. После чего все формулы, содержащие внешние ссылки, будут обновлены.

Если в рабочей книге содержались внешние ссылки, которые впоследствии будут не нужны, можно преобразовать их в значения, разорвав все связи. Для этого необходимо выполнить следующие действия:

1. Выбрать диапазон, в котором находятся формулы с внешними ссылками, и скопировать его в буфер обмена.
2. Выполнить команду *Правка* → *Специальная вставка*.
3. В диалоговом окне *Специальная вставка* установить переключатель *Значения* (см. рис. 1.18).
4. Нажать кнопку *OK*, после чего все формулы из выбранного диапазона будут заменены их текущими значениями.

1.7.6. Понятие консолидации

Консолидация (consolidation) – это процесс комбинирования или накопления информации из двух или более рабочих таблиц, которые могут находиться в нескольких рабочих книгах.

Под консолидацией подразумевается ряд операций по объединению данных нескольких рабочих таблиц или рабочих книг с использованием математических функций (сумма, произведение, среднее значение, минимальное значение и т. д.).

Пример 1

Бюджеты всех отделов предприятия хранятся в отдельных рабочих таблицах одной рабочей книги. Необходимо консолидировать (объединить) эти данные и создать итоговый документ бюджета всего предприятия.

Пример 2

Все руководители отделов представляют свои бюджеты в отдельных рабочих книгах. Необходимо консолидировать эти файлы в общий бюджет предприятия.

Существует несколько способов (методов) консолидации данных:

1. Использование формул, содержащих внешние ссылки.
2. Копирование данных и применение команды *Специальная вставка*.
3. Использование команды *Данные → Консолидация*.

1.7.7. Консолидация рабочих таблиц с помощью формул

Этот способ предполагает создание формул, содержащих ссылки на другие рабочие таблицы или рабочие книги. Преимуществом метода является динамическое обновление данных, т. е. в случае изменения значений в исходной рабочей книге формулы пересчитываются автоматически. Во время создания консолидирующих формул исходные рабочие книги необязательно должны быть открыты.

Пример 1

Вычислить сумму значений из ячеек A1, находящихся в рабочих таблицах, начиная с листа Лист2 и заканчивая листом Лист10.

Решение

=СУММ(Лист2:Лист10!A1).

Если консолидация затрагивает другие рабочие книги, то можно воспользоваться формулами, содержащими внешние ссылки:

Пример 2

=[Регион1.xls]Лист1!A1+[Регион2.xls]Лист1!A1+[Регион3.xls]Лист1!A1.

1.7.8. Использование команды «Специальная вставка»

Этот метод применим только в том случае, если все рабочие таблицы, которые консолидируются, открыты. Недостатком метода является то, что консолидация получается нединамичной, т. е. при этом не создается формула. Поэтому, если любые консолидированные данные изменятся, то результат станет неправильным. Преимуществом является то, что при вставке данных из буфера обмена, над ними можно выполнять математические операции.

При использовании метода необходимо выполнить следующие действия:

1. Скопировать данные из первого диапазона исходной таблицы.
2. Активизировать зависимую рабочую книгу и выбрать ячейку, в которую необходимо поместить консолидированные данные.
3. Выполнить команду *Правка → Специальная вставка*.
4. Установить переключатель *Сложить* (см. рис. 1.18).

1.7.9. Использование команды «Данные → Консолидация»

Этот метод является универсальным и применяется в случае, когда способы размещения информации в исходных рабочих таблицах различны. С помощью метода можно выполнять статичную консолидацию (без формул связи) и динамичную консолидацию (с формулами связи).

Команда *Данные → Консолидация* поддерживает следующие методы консолидации:

- *по позиции* (используется в том случае, если способы информации в исходных рабочих книгах идентичны);
- *по категориям* (использует возможность подбора данных из исходных рабочих таблиц по заголовкам строк и столбцов. Можно использовать, если данные в исходных рабочих таблицах размещены без соответствия).

При использовании метода необходимо выполнить следующие действия:

1. Выполнить команду *Данные → Консолидация*. На экране появится диалоговое окно *Консолидация*, приведенное на рис. 1.20.

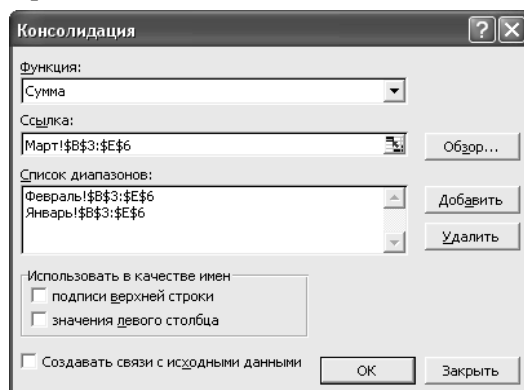


Рис. 1.20. Диалоговое окно *Связи*

2. Выбрать необходимый тип консолидации (например, *Сумма*).
3. Если рабочая книга открыта, ввести ссылку на диапазон первой рабочей таблицы из тех, которые должны быть консолидированы. Если рабочая книга закрыта, то найти и открыть ее с помощью кнопки *Обзор*.
4. Ввести ссылки на диапазоны других рабочих таблиц, которые должны быть консолидированы, и выбрать кнопку *Добавить*. Ссылки будут добавлены к списку диапазонов.
5. Поскольку способы размещения информации в рабочих таблицах различны, необходимо выбрать опции *В левом столбце* и *В верхней строке*. В результате Excel будет подбирать данные по заголовкам.
6. Выбрать опцию *Создавать связи с исходными данными*. В результате Excel создаст структуру, содержащую внешние ссылки. Если опция не будет выбрана, консолидация окажется статической и при изменении данных рабочих таблиц консолидация не будет обновлена автоматически.
7. Нажать *ОК* для выполнения консолидации.

1.8. Создание и использование формул массивов

1.8.1. Понятие формулы массива

Массив – это набор ячеек или значений, которые обрабатываются как одна группа. Элементы массива могут содержаться в группе ячеек или быть поименованной константой.

Формула массива – формула, в которой используется один или несколько массивов непосредственно или в качестве аргументов функций. Формула массива может занимать одну или несколько ячеек, т. е. формула массива – это формула специального вида, которая работает с группами ячеек. Формулу массива всегда можно представить как простую формулу, которая использует несколько значений и возвращает несколько результатов, причем каждый результат помещается в отдельную ячейку.

В Excel массивы могут быть одно- или двумерными. Измерения массивов непосредственно соответствуют строкам и столбцам. Например, одномерный массив может быть группой ячеек, которые размещены в одной строке (горизонтальный массив) или в одном столбце (вертикальный массив). Двумерный массив размещается в одной или более строках и столбцах.

Операции над массивами производятся с помощью формул массивов. Эти формулы вводятся с помощью комбинации клавиш *Ctrl+Shift+Enter*, в то время как обычная формула вводится с помощью клавиши *Enter*.

Пример

В диапазонах A1:A5 и B1:B5 введены пятиэлементные вертикальные массивы. Нужно найти произведение этих массивов в столбце C. Расчетная таблица приведена на рис. 1.21.

	A	B	C
1	3	5	15
2	2	6	12
3	1	8	8
4	4	3	12
5	5	5	25

Рис. 1.21. Расчетная таблица примера о произведении массивов

Решение

Для ввода формулы массива необходимо:

1. Отметить итоговый диапазон C1:C5.
2. Ввести в итоговый диапазон формулу $=A1:A5*B1:B5$.
3. Нажать *Ctrl+Shift+Enter* для того, чтобы формула стала формулой массива и к ней добавились фигурные скобки $\{=A1:A5*B1:B5\}$.

В результате получился также пятиэлементный вертикальный массив, каждый элемент которого состоит из произведения соответствующих элементов первого и второго массивов. Поскольку Excel может разместить в отдельной ячейке только одно значение, результат операции будет занимать пять ячеек, для каждой из которой используется одна и та же формула массива: $\{=A1:A5*B1:B5\}$.

Для редактирования формулы массива необходимо отметить все ячейки группы массива, активизировать строку формул и нажать клавишу *F2*. При этом Excel удаляет фигурные скобки. Закончив редактирование, необходимо снова нажать *Ctrl+Shift+Enter*. Если формула массива помещена в несколько ячеек, то при редактировании необходимо использовать всю группу как одну ячейку.

При использовании формул массивов запрещено следующее:

- изменять содержимое одной из ячеек, входящих в формулу массива;

- перемещать отдельные ячейки, на которые распространяется формула массива (можно перемещать все ячейки формулы массива одновременно);
- удалять отдельные ячейки, на которые распространяется формула массива (можно удалить только весь массив целиком);
- вставлять новые ячейки в массив, а также новые строки и столбцы, которые добавляют новые ячейки к группе массива.

1.8.2. Использование массивов констант

В формулах массивов можно использовать не только группы ячеек, а также массивы констант. Его можно ввести непосредственно в виде формулы или определить с помощью диалогового окна *Присвоение имени* (*Вставка* → *Имя* → *Присвоить*), приведенного на рис. 1.22, предварительно выделив необходимый диапазон ячеек. Массивы констант могут быть использованы в формулах массивов вместо ссылки на диапазон ячеек. Для того, чтобы использовать массив констант, необходимо в формуле массива ввести набор значений и заключить его в фигурные скобки. Можно также воспользоваться именем массива, если оно было предварительно присвоено.

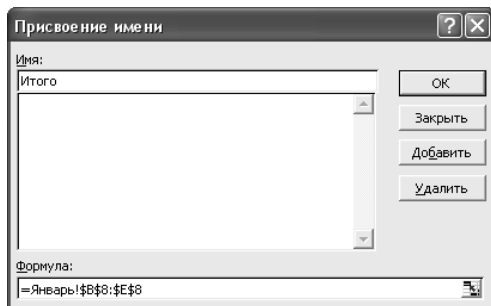


Рис. 1.22. Диалоговое окно *Присвоение имени*

Массив констант может быть одномерным и двумерным. Одномерные массивы подразделяются на вертикальные и горизонтальные. Элементы одномерного горизонтального массива отделяются точкой с запятой.

Пример 1

{1;2;3;4;5}. Для ввода этого массива в диапазон ячеек необходимо отметить по горизонтали пять ячеек, затем ввести {1;2;3;4;5} и нажать *Ctrl+Shift+Enter*.

Элементы одномерного вертикального массива отделяются двоеточием.

Пример 2

{10:20:30:40:50:60}. Для ввода этого массива в диапазон ячеек необходимо отметить по вертикали шесть ячеек, затем ввести {10:20:30:40:50:60} и нажать *Ctrl+Shift+Enter*.

В двумерном массиве элементы одной строки также отделяются точкой с запятой, а строки отделяются двоеточием.

Пример 3

{1;2;3;4;5;6;7;8;9;10;11;12} – это массив размерностью 3×4 (3 строки, каждая из которых занимает 4 ячейки).

В формуле массива нельзя использовать список ссылок на ячейки, список имен или формул аналогично тому, как это задается в массиве констант. Также важно отслеживать размерности массивов.

Пример 4

Запись {2*3;3*4;4*3} неверна, так как в ней содержатся формулы.

Пример 5

Запись {A1;B1;C1} неверна, так как в ней использованы ссылки на ячейки. Необходимо использовать ссылку на диапазон {A1:C1}.

Пример 6

Формула {2;3;4}*{10;11} неверна и выдаст значения 20;33 и #Н/Д, так как размерности массивов не совпадают.

1.8.3. Примеры использования формул массивов

Пример 1

Даны выборки по двум измерениям. Необходимо вычислить среднее отклонение по выборкам с помощью формулы массива (рис. 1.23).

Решение

В ячейку C9 нужно ввести формулу массивов: {=СРЗНАЧ(C2:C8–B2:B8)}.

	А	В	С
1	Выборка	Измерение 1	Измерение 2
2	Выборка1	84	87
3	Выборка2	75	73
4	Выборка3	84	85
5	Выборка4	88	92
6	Выборка5	93	93
7	Выборка6	84	91
8	Выборка7	90	93
9	Среднее отклонение		2,285714286

Рис. 1.23. Расчетная таблица примера 1

Пример 2

В диапазон А3:А6 введены числа: 6, 8, 9, 12 (рис. 1.24). Нужно вычислить квадратные корни этих чисел и поместить их соответственно в диапазон В3:В6, используя обычные формулы и формулы массива.

Решение

Результат представлен в столбце В на рис. 1.24, а расчетные формулы – в столбцах С и D.

	А	В	С	Д
1	Вычисление квадратного корня			
2	Заданное число	Квадратный корень	Обычная формула	Формула массива
3	6	2,44949	=КОРЕНЬ(А3)	{=КОРЕНЬ(А3:А6)}
4	8	2,828427	=КОРЕНЬ(А4)	{=КОРЕНЬ(А3:А6)}
5	9	3	=КОРЕНЬ(А5)	{=КОРЕНЬ(А3:А6)}
6	12	3,464102	=КОРЕНЬ(А6)	{=КОРЕНЬ(А3:А6)}

Рис. 1.24. Расчетная таблица примера 2

Пример 3

В диапазон А3:А11 введены данные, в которых содержатся положительные и отрицательные числа. Необходимо вычислить среднее из положительных чисел в этой группе.

Решение

Для нахождения среднего из положительных чисел можно использовать:

1. Функции СУММЕСЛИ и СЧЕТЕСЛИ

=СУММЕСЛИ(А3:А11; ">0"; А3:А11)/СЧЕТЕСЛИ(А3:А11; ">0").

2. Формулу массивов {=СРЗНАЧ(ЕСЛИ(А3:А11>0; А3: А11; ""))}.

1.9. Форматы данных, поддерживаемые Excel

Данные – это способ представления информации, которая может поступать в любой форме и из различных источников. К форматам данных, поддерживаемых Excel относятся следующие:

- *текстовые файлы* (в качестве разделителей полей используются пробелы, символы табуляции и запятые);
- *файлы Lotus 1-2-3* – файлы электронных таблиц, созданные в табличном редакторе Lotus версий 1-2-3 для Windows;
- *файлы Quattro Pro/DOS* – файлы электронных таблиц, созданные в табличном редакторе Quattro Pro для DOS фирмы Novell;
- *файлы Microsoft Works 2.0* – файлы, созданные в интегрированном пакете Microsoft Works 2.0;
- *файлы dBase* – файлы баз данных в формате DBF;
- *файлы SYLK* – файлы таблиц, созданные в программе MultiPlan фирмы Microsoft;
- *Web-страницы* – файлы формата HTML;
- *файлы Query* – файлы, содержащие запросы на извлечение информации из внешних баз данных.

Excel содержит четыре *дополнительных формата*, предназначенных для работы с базами данных и списками адресов:

- почтовый индекс;
- индекс+4;
- номер телефона;
- табельный номер.

Числа, введенные в этих форматах, в действительности остаются числами, а не текстовыми значениями при наличии круглых скобок и тире. Форматы устанавливаются в диалоговом окне *Формат ячеек* на вкладке *Число*. Если пользователю необходимо установить свой личный формат, можно использовать категорию *Все форматы* окна *Формат ячеек* вкладки *Число*.

Пользовательский формат может содержать до четырех секций кодов, отделенных точками с запятой, и записывается в поле *Тип*. Интерпретация пользовательских форматов приведена в табл. 1.4.

Таблица 1.4. Интерпретация пользовательских форматов

Код	Назначение
0	Метка цифры гарантирует вывод заданного количества цифр. Если нужно, добавляет незначащие нули Например, формат 0,00 выведет число ,987 как 0,99
#	Аналогичен символу 0, однако незначащие нули не выводятся. Для вывода нуля можно задать формат #,###0
?	Метка цифры. Вместо незначащих нулей ставится пробел. Такая метка позволяет выравнивать числа по десятичной запятой Например, 1,4 и 4,45 будут выровнены по десятичной запятой, если задать формат 0,??
,	Задаёт положение десятичной запятой
%	Символ процентного формата. Число умножается на 100 и заканчивается символом %
/	Символ дробного формата. Выводит дробную часть в виде простой дроби Например, формат #/?/? выводит дробь 0,269 как ¼ А формат #??/?/? выводит дробь 0,269 как 46/171
Пробел	Разделитель групп разрядов
\$ - + / ()	Стандартные символы форматирования. Обратная косая черта при выводе пропускается Например, формат # ## " "\P; -# ## " "\O выведет положительные числа с пробелом и буквой П, а отрицательные – с минусом, пробелом и буквой О
_	Подчеркивание. Вставляет пробел, равный ширине следующего символа
"Текст"	Кавычки. Отражают весь текст, заключенный в кавычки
*	Символ повторения. Следующий за * символ повторяется до конца ширины столбца
@	Метка текста. Если ячейка содержит текст, то в отображаемом значении этот текст появится в той позиции, где в коде формата находится символ @

Пример 1

Требуется задать формат для ввода даты: название дня недели, полное название месяца, число и год.

Решение

ДДДД, ММММ ДД, ГГГГ.

Пример 2

Требуется задать формат для инвентарных номеров таким образом, чтобы любое число, например, 1234567, преобразовывалось в Инв. № 123-4567.

Решение

"Инв. № "####-####.

1.10. Импортрование данных

1.10.1. Обзор методов импортирования

Существует несколько способов импортирования данных в рабочую таблицу Excel:

- генерирование данных с помощью использования формул и макросов;
- копирование данных из другого приложения с использованием буфера обмена;
- импортрование данных из файла, созданного в другом приложении;
- использование надстройки MS Query для перенесения данных из внешних баз данных.

Далее рассматривается импортрование данных с использованием буфера обмена, из других приложений, с помощью надстройки MS Query.

1.10.2. Использование буфера обмена для ввода данных

Буфер обмена – это область памяти компьютера, используемая как место хранения общей информации, которая вырезана или скопирована из какого-либо приложения. Данные, полученные из буфера обмена, остаются в нем после вставки и могут использоваться несколько раз. Так как буфер обмена может содержать только одну запись, то при копировании и вырезании чего-либо нового, старое содержимое стирается. В зависимости от типа данных, которые копировались, и от форматов, поддерживаемых буфером обмена, полученный результат может отличаться от оригинала.

В инсталляцию Windows включена специальная программа просмотра буфера обмена Clipboard Viewer,

которая показывает его содержимое. Она находится в папке Windows.

При копировании или вырезании данных из буфера обмена, приложение-источник помещает их в одном или нескольких форматах. Различные приложения поддерживают разные форматы буфера обмена. При вставке данных в другое приложение, приложение-приемник определяет, в каком формате принимать данные. Нужный формат обычно выбирается таким образом, чтобы сохранить основную информацию и положение места вставки. В некоторых случаях для отображения данных, находящихся в буфере обмена, в другом формате можно использовать команду *Вид* программы Clipboard Viewer. Например, если в буфер обмена скопирован диапазон ячеек, то их можно представить в виде векторного рисунка, растрового изображения, текста и в других форматах.

Форматы представления данных, находящихся в буфере обмена, не влияют на то, как копируются данные. Однако, в некоторых случаях для выбора альтернативного метода вставки данных удобно использовать команду *Правка* → *Специальная вставка*.

1.10.3. Импортирование данных из файла, созданного в другом приложении

В Excel реализована возможность импортирования данных, находящихся в других приложениях. К таким данным относятся: текстовые файлы, файлы MSQuery, Web-страницы, файлы других табличных процессоров (Lotus 1-2-3, Quattro Pro, более ранних версий Excel). Форматы этих данных можно установить при открытии файла в Excel, указав в диалоговом окне *Открытие документа* необходимый тип файла.

Импортирование текстовых файлов

Текстовые файлы очень распространены для импортирования в Excel. Они содержат только данные без всяких атрибутов форматирования, поэтому многие приложения могут читать и сохранять их. Кроме того, эти файлы удастся использовать для переноса данных между приложениями, не поддерживающими общепринятый формат.

С точки зрения табличной базы данных, каждая строка в текстовом файле соответствует одной записи базы данных и каждая запись состоит из некоторого количества полей. В Excel каждая строка (или запись) импортируется в отдельную строку листа рабочей книги, а каждое поле – в отдельный столбец.

Текстовые файлы бывают двух типов:

- размеченные;
- неразмеченные.

Размеченные файлы состоят собственно из текста и символов конца строки. При разметке текста для отделения полей в одной строке используют специальные символы. Обычно в качестве этих символов используют запятую, точку с запятой, пробел или символ табуляции (иногда используют другие разделители).

Неразмеченный файл не имеет специальных разделителей полей. Однако часто поля имеют фиксированную длину, что значительно облегчает процедуру разбивки каждой строки текста на столбцы.

Для импортирования текстовых файлов используют *Мастер текстов*. Он загружается при выборе в диалоговом окне *Открытие документа* текстового файла (рис. 1.25). *Мастер текстов* представляет собой серию интерактивных окон, в которых определяется информация для разбиения строк текстового файла на столбцы электронной таблицы.

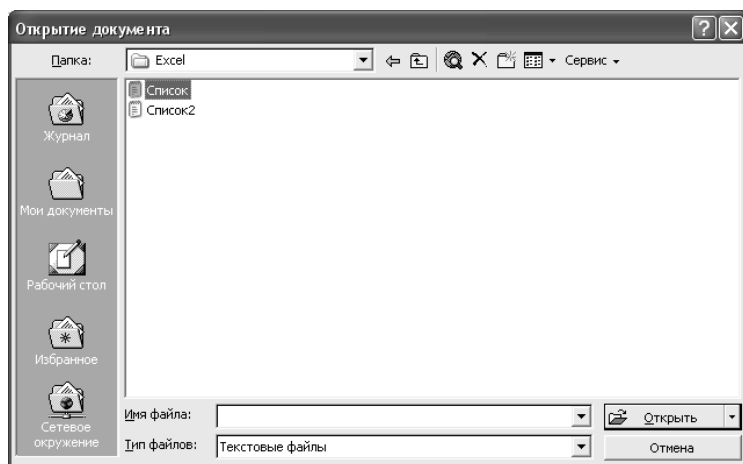


Рис. 1.25. Настройка диалогового окна *Открытие документа* для запуска *Мастера текстов*

Мастер текстов включает три шага.

Шаг 1 (рис. 1.26). Мастер текстов проверяет тип импортируемого текстового файла (список *Формат файла*), позволяет выбирать строку, с которой начинается импортирование (можно пропускать строку заголовка). Встроенное окно предварительного просмотра показывает используемый набор символов.

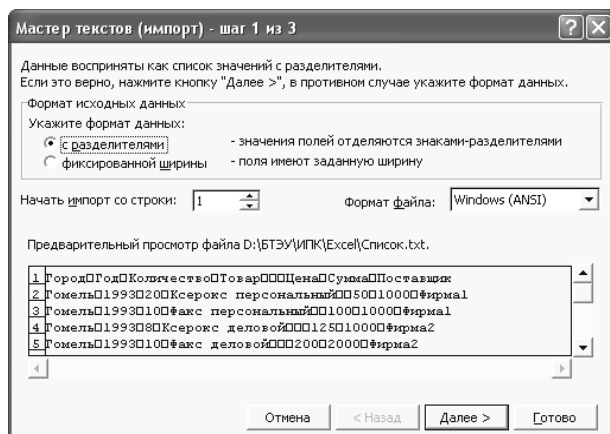


Рис. 1.26. Первое диалоговое окно Мастера текстов

Шаг 2 (рис. 1.27). Диалоговые окна этого шага могут быть различными в зависимости от выбранного формата данных. В них можно определить тип разделителя, символ-ограничитель строк текста, а также произвести разбиение на столбцы.

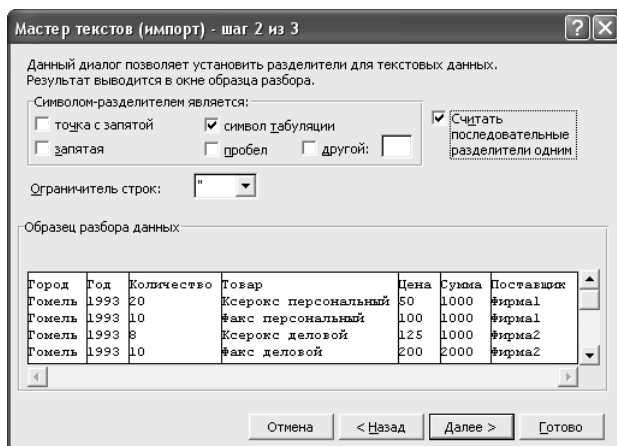


Рис. 1.27. Второе диалоговое окно Мастера текстов

Шаг 3 (рис. 1.28). Мастер текстов позволяет определить форматы данных относительно каждого столбца. Наиболее универсальным является *общий формат*. Для значений этого формата осуществляется автоматическое преобразование числовых значений в числа, дат – в даты, всех прочих – в текст.

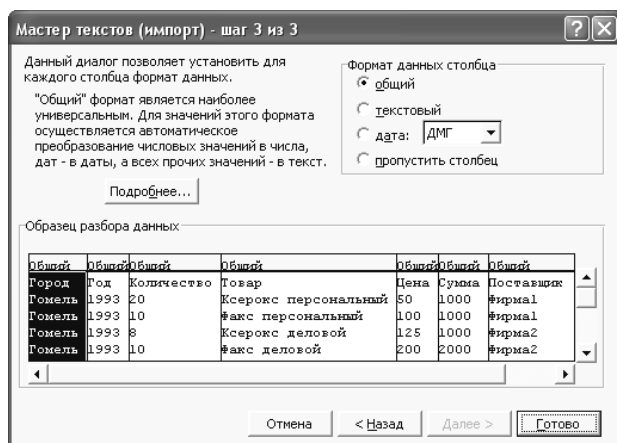


Рис. 1.28. Третье диалоговое окно Мастера текстов

При работе с данными некоторые пользователи предпочитают использовать возможности доступа к данным, хранимым в файлах внешних баз данных. Для импортирования этих данных необходимо сделать запрос (query) в базу данных и загрузить в лист рабочей таблицы ту часть внешней базы данных, которая будет использоваться.

Для работы с файлами внешних баз данных существует специальное приложение MS Query, входящее в комплект надстроек Excel. Оно предназначено для подключения к внешним источникам данных и может использоваться в следующих случаях:

- файл данных слишком большой (MS Query размещает в оперативной памяти только те данные исходного файла, которые удовлетворяют заданным критериям);
- с файлом данных одновременно работают несколько пользователей (подключение к файлу через MS Query позволяет обновить лист Excel при внесении изменений в базу данных другими пользователями);
- необходимо объединить несколько таблиц данных (MS Query, в отличие от Excel, позволяет объединять таблицы, которые имеют общие поля).

Для того, чтобы определить, установлено ли приложение MS Query на компьютере, необходимо в Excel выбрать команду *Данные*. Если в появившемся меню доступна команда *Внешние данные* → *Создать запрос*, то MS Query установлено. В противном случае необходимо выбрать команду *Сервис* → *Надстройки* → *Настройка MS Query для Excel* → *Ok*.

Схема получения внешних данных по запросу приведена на рисунках 1.29–1.33. Вначале необходимо выбрать тип СУБД, затем в диалоговом окне *Выбор базы данных* указать место нахождения базы данных, из которой будут импортироваться данные. В окне *Создание запроса: выбор столбцов* выбрать источники для создания запроса, а затем отобразить записи, используя критерии отбора (окно *Создание запроса: отбор данных*). После выполнения команды *Вернуть данные в Excel*, выбранные данные скопируются на лист рабочей таблицы, где их можно обрабатывать по своему усмотрению.

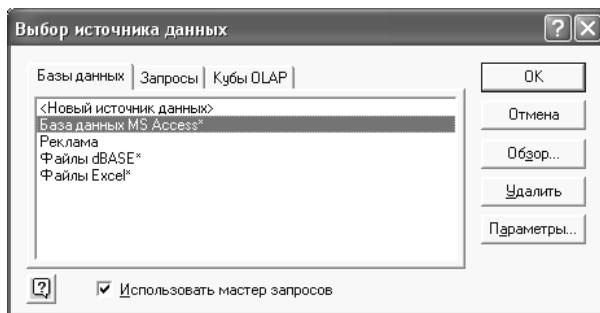


Рис. 1.29. Диалоговое окно *Выбор источника данных*

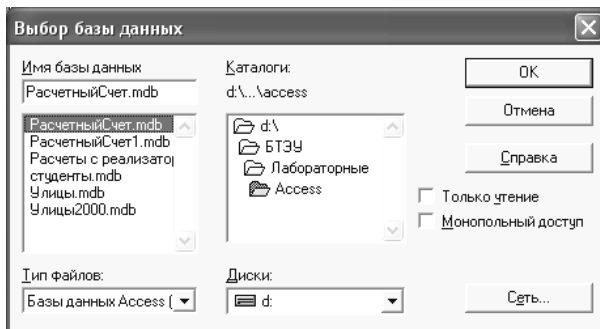


Рис. 1.30. Диалоговое окно *Выбор базы данных*

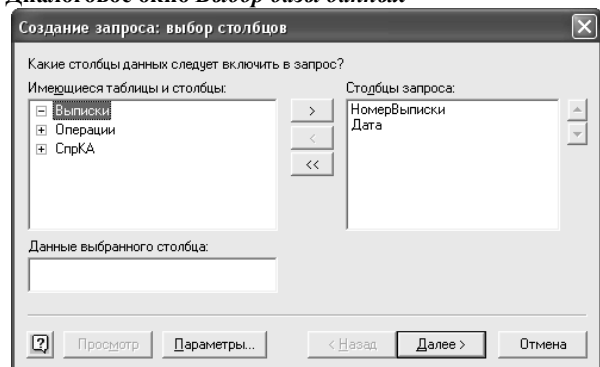


Рис. 1.31. Диалоговое окно *Создание запроса: выбор столбцов*

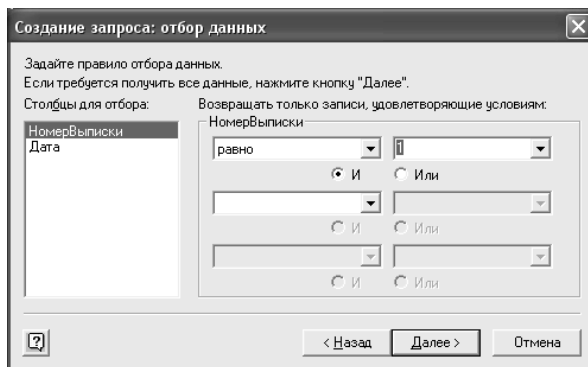


Рис. 1.32. Диалоговое окно *Создание запроса: отбор данных*

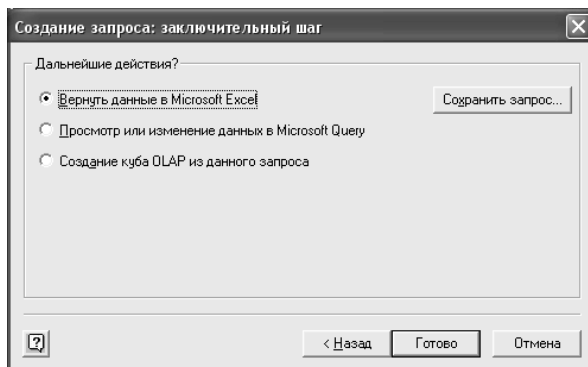


Рис. 1.33. Заключительное диалоговое окно создания запроса

1.11. Работа со списками

1.11.1. Понятие списка

Для обозначения базы данных, хранящейся на листе рабочей таблицы, используется термин *список*. Для таблиц с информацией, находящихся во внешних файлах специального формата, используется термин *база данных*. Таким образом, список можно считать табличной базой данных, которая представляет собой непрерывный прямоугольный диапазон ячеек рабочего листа, содержащий данные определенной структуры.

Список – это упорядоченный набор данных в виде прямоугольной таблицы, характеризующий множество некоторых однотипных объектов. Обычно список состоит из строки заголовков (текст описания) и дополнительных строк данных, которые могут быть числовыми или текстовыми. Пример списка приведен на рис. 1.34.

	А	В	С	Д	Е	Ф	Г
	Область	Город	Номер магазина	Дата продажи	Наименование товара	Цена единицы товара	Количество единиц товара
1							
2	Брестская	Брест	1	14.03.2006	сок	1550	340
3	Витебская	Полоцк	2	14.03.2006	сахар	1320	450
4	Гомельская	Гомель	5	12.03.2006	сок	1600	670
5	Гомельская	Гомель	8	15.03.2006	колбаса копченая	18700	370
6	Гомельская	Речица	1	14.03.2006	сосиски	8800	580
7	Гомельская	Жлобин	2	11.03.2006	шоколад	1500	210
8	Витебская	Витебск	2	16.03.2006	сахар	1330	380
9	Витебская	Новополоцк	4	17.03.2006	сахар	1320	460
10	Гродненская	Гродно	7	15.03.2006	шоколад	1450	150

Рис. 1.34. Пример списка

Столбцы списка называют *полями*, а строки – *записями*. Количество столбцов является постоянным, а количество строк – переменным, что позволяет добавлять, удалять или обновлять записи для того, чтобы список содержал оперативную информацию.

1.11.2. Планирование списка

Перед созданием списка необходимо продумать его структуру, т. е. какие данные должен содержать список и какие столбцы следует создать. При планировании списка следует выполнять ряд рекомендаций:

- заголовки полей размещать в первой строке списка, выбирая краткие названия;
- каждый столбец должен содержать однотипную информацию;
- допустимо работать с формулами, которые используют значения из других полей этой же записи. Однако, если формула ссылается на ячейку, расположенную вне списка, то ссылка на эту ячейку должна быть абсолютной, иначе результаты при сортировке списка могут быть неверными;
- не использовать пустых строк и столбцов в списке, так как при проведении операций над списком Excel определяет его границы автоматически, и пустая строка или столбец будут означать конец списка;
- помещать список на отдельном листе. Если это невозможно, то список должен быть отделен от других данных рабочего листа хотя бы одним пустым столбцом или пустой строкой;
- использовать команду *Окно → Закрепить области*, для того, чтобы заголовки были всегда видны при прокручивании листа списка (предварительно надо активизировать левую верхнюю ячейку прокручиваемой области);
- отформатировать полностью весь столбец, чтобы данные впоследствии всегда имели один и тот же формат.

1.11.3. Операции со списками

Над списками Excel можно производить следующие *операции*:

- ввод данных в список;
- фильтрация списков, которая выполняется в целях выборочного отображения строк по определенному критерию;
- сортировка списка;
- вставка формул для подведения промежуточных итогов;
- создание формул для вычисления результатов в списке, отфильтрованном по определенному критерию;
- создание итоговой таблицы данных списка с помощью сводной таблицы.

1.11.4. Ввод данных в список

Данные в список можно ввести тремя способами:

- вручную, используя стандартные методы ввода данных;
- импортировать или скопировать данные из другого файла;
- использовать диалоговое окно форм ввода.

Для *первого способа* ввода данных разработаны инструменты, которые помогают избежать выполнения однообразной операции ввода данных. Один из инструментов – *Автозаполнение*. При вводе Excel просматривает столбец и, если находит закономерность, то заполняет остаток ячеек автоматически. Чтобы завершить ввод данных, надо нажать клавишу *Enter*. Второй из инструментов называется *Выбор из списка*, который находится в контекстном меню вводимой ячейки.

Второй способ предназначен для данных, находящихся во внешнем файле произвольного формата. Для доступа к таким данным предназначен рассмотренный выше *Мастер текстов*.

Третий способ реализуется выполнением команды *Данные → Форма*. На экран выводится диалоговое окно для ввода данных, приведенное на рис. 1.35. Предварительно курсор необходимо установить в строку новой записи. Excel определяет размер списка и в диалоговом окне будет отображать каждое поле списка. Поля, содержащие формулу, не помещаются в окно для редактирования. Для добавления новой записи используется кнопка *Добавить*.

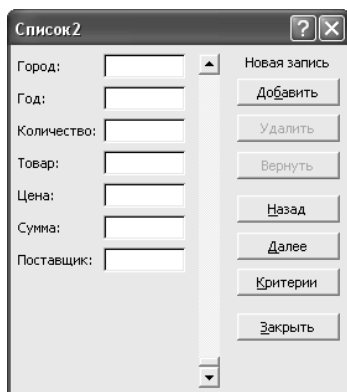


Рис. 1.35. Пример диалогового окна *Форма*

Диалоговое окно, предназначенное для ввода данных, можно также использовать для редактирования

существующих записей, просмотра и удаления записей, выборочного отображения записей по определенному критерию. Поэтому данное окно содержит ряд дополнительных кнопок:

- *Удалить* удаляет текущую запись.
- *Вернуть* отменяет изменения в текущей записи. Работает до тех пор, пока не применили кнопку *Добавить*.
- *Назад* осуществляет переход к предыдущей записи списка. Если установлен критерий отбора, то будет сделан переход к предыдущей записи, удовлетворяющей данному критерию.
- *Далее* осуществляет переход к следующей записи списка. Если установлен критерий отбора, то будет сделан переход к следующей записи, удовлетворяющей данному критерию.
- *Критерии* очищает поля и позволяет ввести критерий, по которому будут отбираться записи.
- *Закрыть* закрывает диалоговое окно и записывает введенные данные в рабочую таблицу.

1.11.5. Фильтрация списка

Фильтрация списка – это процесс сокрытия всех строк, кроме тех, которые удовлетворяют определенным критериям. Например, список расписания занятий можно отфильтровать так, чтобы видеть только записи с одной дисциплиной.

Списки Excel можно фильтровать двумя способами:

- автоматическая фильтрация (автофильтр) – для фильтрации по простым критериям;
- расширенная фильтрация – для фильтрации по сложным (произвольным) критериям.

Автоматическая фильтрация

Для того, чтобы автоматически отфильтровать список, необходимо сначала установить курсор на одну из его ячеек. Затем выполнить команду *Данные → Фильтр → Автофильтр*. Excel проанализирует список и добавит в строку заголовков поля управления – раскрывающиеся списки, как показано на рис. 1.36. В каждом элементе собраны уникальные значения, расположенные в этом столбце. Выбор какого-либо значения приводит к сокрытию всех строк с другими значениями. Список можно одновременно фильтровать по нескольким столбцам.

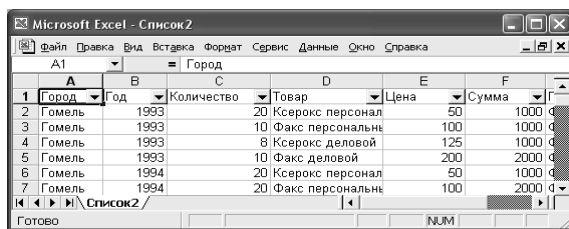


Рис. 1.36. Пример использования автофильтра

После того как список отфильтрован, в строке состояния появится сообщение о том, сколько строк отобрано. Кроме того, изменится цвет стрелки раскрывающегося списка. Это будет служить напоминанием о том, что список отфильтрован по значениям, содержащимся в этом столбце.

Автоматическая фильтрация имеет ограничения. В раскрывающемся списке появляются только первые 250 уникальных элементов. Если число элементов превышает указанный предел, то можно использовать средства расширенной фильтрации.

Кроме всех значений столбца, раскрывающийся список содержит следующие элементы:

- *Все* показывает все элементы столбца. Используется для отмены фильтрации столбца.
- *Первые 10* выбирает 10 (или другое количество) наибольших или наименьших элементов списка.
- *Условие* позволяет фильтровать список по нескольким условиям.
- *Пустые* фильтрует список, показывая только строки с пустыми ячейками в столбце.
- *Непустые* фильтрует список, показывая только строки с непустыми ячейками в столбце.

Если в таблице есть формулы, в которых используются ссылки на ячейки отфильтрованного списка, то после выполнения фильтрации значение, вычисляемое по таким формулам, не изменяется. Например, если ячейка содержит формулу суммирования значений в столбце С, то после фильтрации будет продолжаться суммирование всех значений столбца С, а не только элементов в видимых строках.

Если пользователю необходимо фильтровать списки с использованием нескольких критериев, используется элемент раскрывающегося списка *Условие*. Появившееся диалоговое окно, приведенное на рис. 1.37, позволяет установить *Пользовательский автофильтр*, который имеет несколько *опций*:

- значения больше или меньше установленного (например, отобрать все записи, указывающие на цены товаров, превышающие 2000 р.);
- значения в интервале (например, отобрать все записи, указывающие на цены товаров, превышающие 1000 р. и не превышающие 5000 р.);
- два отдельных значения (например, отобрать записи, в которых находится информация об объеме

продаж в городах Гомель ИЛИ Брест);

- выборка по шаблону, используя символы подстановки * и ? (например, для того, чтобы вывести на экран записи только о тех товарах, наименование которых начинается с буквы М, используется шаблон М*).

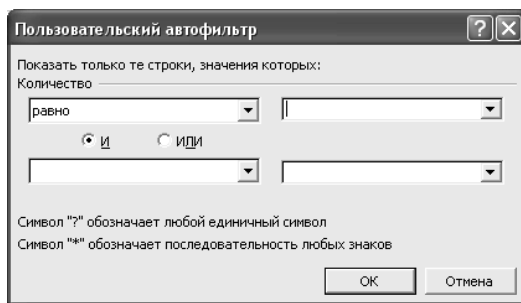


Рис. 1.37. Диалоговое окно *Пользовательский автофильтр*

Расширенная фильтрация

Если недостаточно возможностей автофильтра, используют *расширенную фильтрацию*. Этот фильтр предоставляет следующие возможности:

- определять более сложный критерий фильтрации;
- устанавливать вычисляемый критерий фильтрации;
- перемещать копии строк, отвечающих определенному критерию, в указанную часть таблицы.

Расширенная фильтрация выполняется командой *Данные → Фильтр → Расширенный фильтр*. На экране появляется диалоговое окно, приведенное на рис. 1.38. При этом для использования расширенной фильтрации, кроме исходного диапазона необходимо задать диапазон условий (критериев).

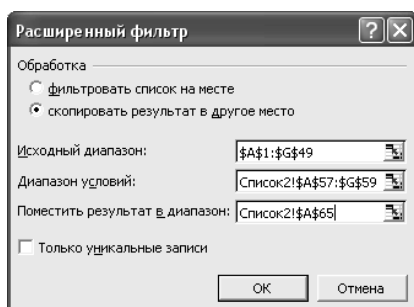


Рис. 1.38. Диалоговое окно *Расширенный фильтр*

Диапазон критериев – это специально отведенная область листа рабочей таблицы, состоящая, по крайней мере, из двух строк, первая из которых должна содержать все или некоторые названия полей списка. Остальные строки должны содержать критерии фильтрации. Удобно размещать диапазон критериев под списком или вообще на другом листе таблицы. Элементы, которые указываются в диапазоне критериев, можно разделить на следующие типы:

1. *Текстовые или числовые критерии* (при фильтрации используется сравнение с числом или строкой с помощью операторов «равно» (=), «больше чем» (>), «меньше чем» (<), «не равно» (<>), «больше или равно» (>=), «меньше или равно» (<=)).

Например:

- отобрать слова, начинающиеся с буквы Л и заканчивающиеся буквой Я (*ответ: >=Л*);
- отобрать слова, кроме тех, которые начинаются с буквы С (*ответ: <>С**);
- отобрать все слова Январь (*ответ: =Январь*);
- отобрать слова, начинающиеся с букв См (*ответ: См**);
- отобрать слова, начинающиеся с буквы С и заканчивающиеся буквой С (*ответ: С*С*);
- отобрать слова размером из трех символов, начинающиеся с буквы С и заканчивающиеся буквой С (*ответ: С?С*).

2. *Вычисляемые критерии* – фильтрация, которая использует любые вычисления. Вычисляемый критерий образует для списка новое поле. При использовании этих критериев следует соблюдать следующие правила:

- не использовать заголовки полей списка в диапазоне критериев, создавать новый заголовок или оставлять пустую ячейку;
- можно использовать любое количество вычисляемых критериев, а также сочетание вычисляемых критериев с невычисляемыми;
- если вычисляемая формула ссылается на значения вне списка, необходимо использовать абсолютные ссылки;

- при создании формул для вычисляемых критериев необходимо использовать относительные ссылки и первую строку списка (не строку заголовков).

1.11.6. Сортировка списка

Сортировка – это упорядочивание записей списка. Для выполнения сортировки необходимо выполнить следующее:

1. Активизировать любую ячейку списка.
2. Выполнить команду *Данные* → *Сортировка*.
3. В открывшемся диалоговом окне *Сортировка диапазона*, приведенном на рис. 1.39, перед указанием ключей сортировки необходимо установить идентификацию полей *По подписям* (по именам полей) или *По обозначениям столбцов листа* (A, B, D и т. д.).
4. В раскрывающемся списке *Сортировать по* (1-й ключ сортировки) выбрать название поля, по которому выполняется сортировка списка, и указать порядок следования записей после сортировки *По возрастанию* или *По убыванию*.
5. Для задания сортировки списка по двум или более полям необходимо выбрать имена полей в раскрывающихся списках *Затем по* (2-й ключ сортировки) и *В последнюю очередь по* (3-й ключ сортировки).
6. Нажать *ОК*.

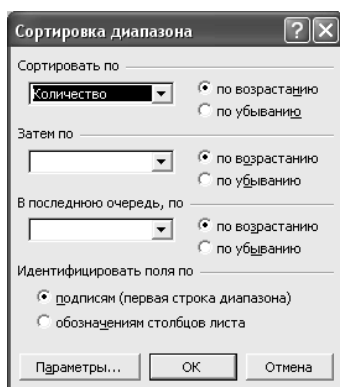


Рис. 1.39. Диалоговое окно *Сортировка диапазона*

При необходимости можно создать свой пользовательский список. Для его создания нужно выполнить команду *Сервис* → *Параметры*. На вкладке *Списки* выбрать опцию *Новый список* и ввести все элементы списка по порядку. По команде *Данные* → *Сортировка* можно отсортировать данные исходного списка по пользовательскому списку.

1.11.7. Создание промежуточных итогов

Создание промежуточных итогов позволяет проанализировать список, вставить формулы промежуточных расчетов и структурировать таблицу. Для подведения промежуточных итогов список должен быть отсортирован, так как итоги будут подводиться каждый раз при изменении значения определенного поля.

Для автоматической вставки формул промежуточных итогов в список необходимо выполнить следующее:

1. Установить табличный курсор в любом месте списка.
2. Выполнить команду *Данные* → *Итоги*.
3. В появившемся диалоговом окне, приведенном на рис. 1.40, задать следующие необходимые значения элементам управления:
 - *При каждом изменении в* показывает все поля списка. Выбранное поле должно быть отсортировано.
 - *Операция* предоставляет на выбор 11 функций. Обычно используется функция *СУММ*, которая задана по умолчанию.
 - *Добавить итоги по* выводит названия всех полей списка, позволяет установить метку поля, по которому необходимо подвести итоги.
 - *Заменить текущие итоги* заменяет существующие формулы итогов на новые.
 - *Конец страницы между группами* устанавливает символ конца страницы после подведения каждого промежуточного итога.
 - *Итоги под данными* располагает итоги под текущими данными. В случае, если опция не отмечена, вначале будет расположен итог, а затем данные.
 - *Убрать все* удаляет все формулы итогов, восстанавливая список.

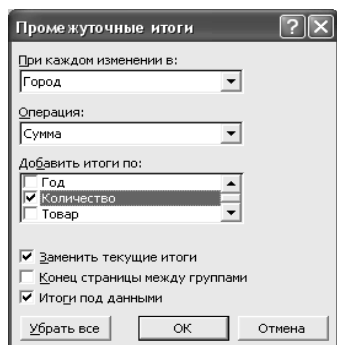


Рис. 1.40. Диалоговое окно *Промежуточные итоги*

Если добавить промежуточные итоги к отфильтрованному списку, то после удаления фильтрации их данные могут стать неверными.

1.12. Анализ данных с помощью сводных таблиц

1.12.1. Понятие и назначение сводных таблиц

Сводные таблицы являются одним из наиболее мощных средств Excel по анализу баз данных, помещенных в таблицы или списки. *Сводная таблица* – это динамический итог данных, содержащихся в списке. Она позволяет создавать динамические перекрестные табличные данные для нескольких измерений данных. С помощью сводной таблицы можно вывести промежуточные и общие итоги с любым необходимым уровнем детализации по нескольким измерениям.

Сводная таблица не просто группирует и обобщает данные, но и дает возможность провести глубокий анализ имеющейся информации. Создавая сводную таблицу, пользователь задает имена полей, которые размещаются в ее строках и столбцах. Допускается также задание поля страницы, которое позволяет работать со сводной таблицей как со стопкой листов.

Таким образом, сводные таблицы удобны по следующим *причинам*:

- позволяют создавать обобщающие таблицы, которые дают возможность группировать однотипные данные, подводить итоги, находить статические характеристики записей;
- позволяют производить автоматический отбор информации;
- на основе сводных таблиц строятся диаграммы, которые динамически перестраиваются вместе с изменениями сводной таблицы.

Сводная таблица включает следующие *элементы*:

- *Данные* – поля, в которых находятся значения.
- *Категория* – поля, которые описывают данные.
- *Группа* – набор элементов, которые обрабатываются как один элемент. Элементы можно объединить в группу вручную или автоматически (например, объединить *даты* в группу *месяцы*).
- *Исходные данные* – данные, которые используются для создания сводной таблицы.
- *Область данных* – ячейки сводной таблицы, в которых содержатся итоговые данные. В Excel предусмотрено несколько способов подведения итогов данных (суммирование, вычисление среднего значения, подсчет количества и др.).

• *Обновить* – заново вычислить сводную таблицу после того, как в исходные данные были внесены изменения.

• *Общие итоги* – строка или столбец, в котором показаны итоги для всех ячеек строки или столбца сводной таблицы. Всегда можно определить способ подведения общих итогов для строк, столбцов, для строк и столбцов одновременно или не подводить итог.

• *Поле столбца* – поле, которому в сводной таблице соответствует отдельный столбец. Каждый элемент, принадлежащий этому полю, занимает один столбец. Поле столбца может быть вложенным.

• *Поле страницы* – поле в сводной таблице, которое имеет ориентацию страницы. В поле страницы может быть отображен лишь один элемент за один раз.

• *Поле строки* – поле, которому в сводной таблице соответствует отдельная строка. Каждый элемент, принадлежащий этому полю, занимает одну строку. Поле строки может быть вложенным.

• *Промежуточные итоги* – строка или столбец, в которых отображаются промежуточные итоги для текущих ячеек строки или столбца сводной таблицы.

• *Элемент* – отдельный элемент в поле, которое является заголовком строки или столбца сводной таблицы.

Таким образом, любая табличная база данных (список) может иметь несколько полей данных и любое количество категорий. При создании сводной таблицы итог подводится по одному или нескольким полям данных. Значения в полях категорий появляются в сводной таблице в виде строк, столбцов или страниц.

1.12.2. Создание и настройка сводной таблицы

Построение сводных таблиц реализуется с помощью *Мастера сводных таблиц*, который вызывается командой *Данные → Сводная таблица*.

В *первом* диалоговом окне, приведенном на рис. 1.41, определяется тип источника данных. Сводная таблица может создаваться на основе данных, находящихся в следующих местах:

- в списке Excel;
- во внешнем источнике данных;
- в нескольких диапазонах консолидации;
- в другой сводной таблице.

Обычно, анализируемые данные хранятся в списке Excel. Он не должен состоять более, чем из 65536 записей и 256 полей. В первой строке списка должны содержаться названия полей. Данные могут состоять из чисел, текста или формул.

Если для создания сводной таблицы используется внешняя база данных, то данные вводятся на лист рабочей таблицы с помощью приложения MS Query. Можно использовать данные SQL-сервера, файлы dBASE и другие источники.

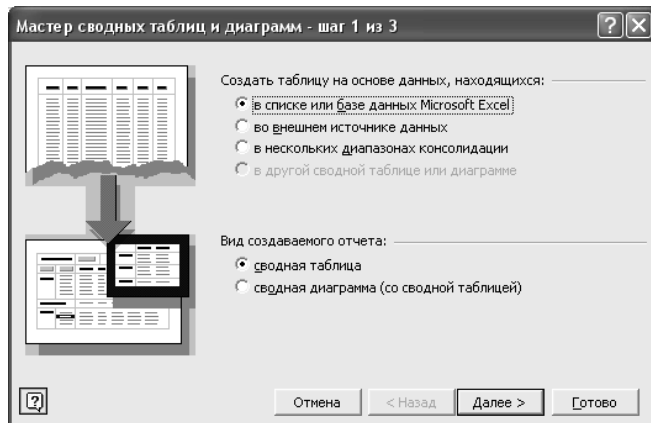


Рис. 1.41. Первое диалоговое окно *Мастера сводных таблиц*

Сводные таблицы можно создавать по информации, находящейся в нескольких рабочих книгах. Эта процедура эквивалентна объединению информации в таблицы. С объединенными данными можно работать, используя все инструменты сводных таблиц.

Excel позволяет создавать сводную таблицу из созданной ранее сводной таблицы. Создаваемая сводная таблица анализирует данные, которые использовались в первой рабочей таблице (но не данные самой сводной таблицы). Если активная рабочая книга не имеет никаких сводных таблиц, то эта опция недоступна.

Во *втором* диалоговом окне (рис. 1.42) *Мастер сводных таблиц* выдает запрос о диапазоне исходных данных. Если при выполнении команды *Данные → Сводная таблица* табличный курсор находился на одном из элементов списка, то Excel автоматически определит диапазон ячеек.

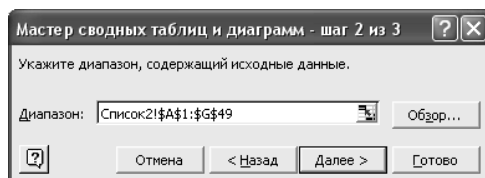


Рис. 1.42. Второе диалоговое окно *Мастера сводных таблиц*

В *третьем* диалоговом окне, приведенном на рис. 1.43, необходимо нажать кнопку *Макет* для непосредственной настройки сводной таблицы. Окно макета, показанное на рис. 1.44, содержит поля списка, представленные в виде столбца с кнопками (правая часть макета) и схему сводной таблицы (левая часть макета).

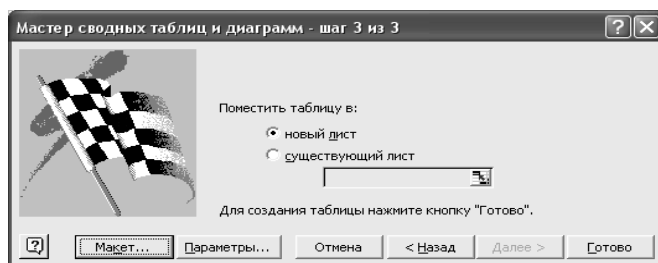


Рис. 1.43. Третье диалоговое окно Мастера сводных таблиц

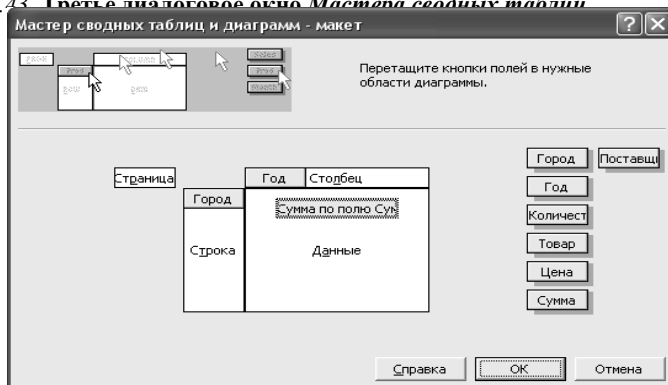


Рис. 1.44. Диалоговое окно макета сводной таблицы

Схема имеет четыре области:

- **Строка** – значения поля используются в качестве заголовков строки в сводной таблице.
- **Столбец** – значения поля используются в качестве заголовков столбцов в сводной таблице.
- **Данные** – поле, для которого подводятся итоги.
- **Страница** – значения поля используются в качестве заголовков страниц в сводной таблице.

В каждую из этих областей можно перетащить любое количество полей из правой части макета. Однако все поля использовать необязательно. Неиспользованные поля не будут присутствовать в сводной таблице. При перетаскивании кнопки поля в область данных *Мастер сводных таблиц* применяет функцию СУММ, если поле содержит числовые значения, и функцию СЧЕТ, если поле содержит нечисловые значения. При настройке сводной таблицы на этом шаге для изменения поля нужно дважды щелкнуть на нем.

При нажатии кнопки *Параметры* в третьем диалоговом окне *Мастера сводных таблиц* появляется диалоговое окно настройки параметров сводной таблицы, приведенное на рис. 1.45. Оно позволяет определить вид сводной таблицы с помощью следующих элементов управления:

- **Имя** определяет имя сводной таблицы, по умолчанию Excel присваивает имена *Сводная таблица1*, *Сводная таблица2*,
- **Общая сумма по столбцам** включает в сводную таблицу общие итоги по столбцам.
- **Общая сумма по строкам** включает в сводную таблицу общие итоги по строкам.
- **Сохранить данные вместе с таблицей** Excel сохраняет дополнительную копию данных, которая называется «кэш сводной таблицы». Это позволяет быстрее пересчитывать сводную таблицу при изменении ее макета.
- **Автоформат** использует автоматическое форматирование даже при изменении макета таблицы.

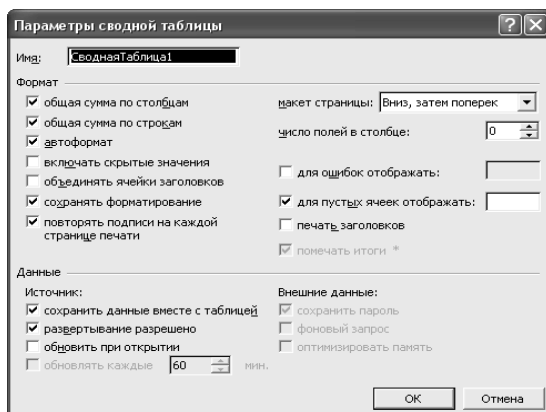


Рис. 1.45. Диалоговое окно параметров сводной таблицы

Пример сводной таблицы, созданной на основе списка, структура которого представлена на рис. 1.36, согласно макету, приведенному на рис. 1.44, показан на рис. 1.46. Она позволяет проанализировать суммарные продажи по годам в разрезе городов. При активизации любой ячейки сводной таблицы появляется панель инструментов *Сводные таблицы*.

The screenshot shows an Excel window titled 'Список2'. The main area contains a PivotTable with 'Город' (City) as the row label and 'Год' (Year) as the column labels. The data is summarized by 'Общий итог' (Grand Total). The task pane on the right, titled 'Сводные таблицы' (PivotTable Tools), shows the 'Сводная таблица' (PivotTable) tab with fields 'Город' and 'Год' assigned to the report filter and columns respectively.

Город	1993	1994	1995	1996	Общий итог
Гомель	5000	6000	7000	8000	26000
Мозырь	4000	3000	2000	4000	13000
Речица	2000	3000	3000	4000	12000
Общий итог	11000	12000	12000	16000	51000

Рис. 1.46. Пример сводной таблицы

1.12.3. Операции со сводными таблицами

Созданная сводная таблица не является статическим объектом. Ее можно изменять и настраивать. При этом возможно изменение структуры таблицы, удаление и добавление полей, обновление и форматирование сводной таблицы, группировка ее элементов.

Обновление сводной таблицы и подведение итогов

Сводная таблица отображает содержание кэш-памяти, а не исходного диапазона данных. Сводная таблица не является динамической таблицей, автоматически обновляемой при изменении данных, на основе которых она была построена. Поэтому, если после создания сводной таблицы изменить данные в рабочей таблице, это не приведет к автоматическому изменению данных в самой сводной таблице. Для ее обновления необходимо активизировать любую ячейку таблицы с данными и выбрать команду *Данные* → *Обновить*. При этом в кэш-память будут вновь занесены сведения, на основе которых строится сводная таблица.

В сводной таблице можно изменять тип вычисляемых промежуточных итогов. Их подводят в том случае, если в строках и столбцах сводной таблицы находится несколько полей. Для подведения итогов необходимо выполнить следующие действия:

1. Активизировать любую внутреннюю ячейку сводной таблицы.
2. Из контекстного меню выполнить команду *Параметры поля*. Появится диалоговое окно *Вычисление поля сводной таблицы*, приведенное на рис. 1.47.

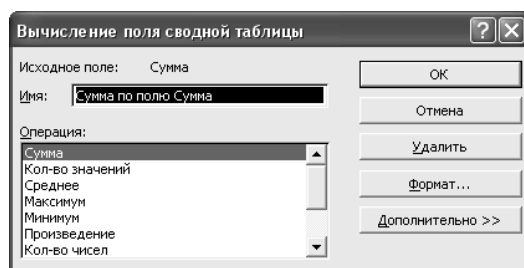


Рис. 1.47. Диалоговое окно *Вычисление поля сводной таблицы* для области данных

3. Из списка *Операция* выбрать одну из допустимых операций (сумма, количество значений, среднее, максимум, минимум, произведение, количество чисел и др.).

4. Если необходимо расширить список допустимых операций, то нужно нажать кнопку *Дополнительно* и выбрать из списка одну из операций для дополнительных вычислений (отличие, доля, с нарастающим итогом, доля от суммы по столбцу и т. п.).

Сортировка данных в сводной таблице

Для того, чтобы упорядочить данные сводной таблицы, используют *сортировку*. Для выполнения сортировки необходимо выполнить следующие действия:

1. Активизировать ячейку из заголовков строк или столбцов сводной таблицы.
2. Из контекстного меню выполнить команду *Параметры поля*. Появится диалоговое окно *Вычисление поля сводной таблицы*, приведенное на рис. 1.48.

3. Нажать кнопку *Далее*. Появится диалоговое окно *Дополнительные параметры поля сводной таблицы*, приведенное на рис. 1.49.

4. Выбрать параметры сортировки (вручную, по возрастанию, по убыванию), а также при необходимости параметры отображения для отбора заданного числа наибольших и наименьших результатов.

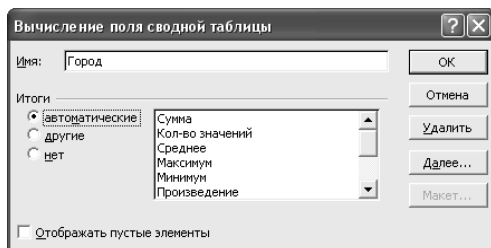


Рис. 1.48. Диалоговое окно *Вычисление поля сводной таблицы* для области заголовков

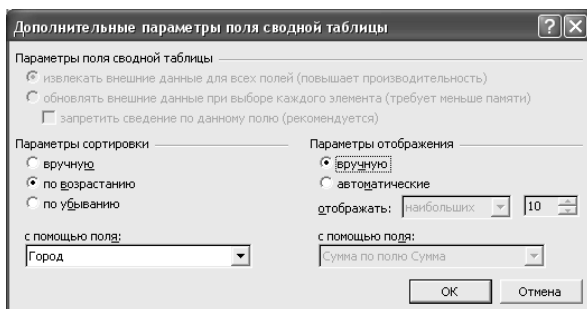


Рис. 1.49. Диалоговое окно дополнительных параметров

Группировка данных в сводной таблице

Сводная таблица позволяет группировать определенные элементы поля. Для этого необходимо выполнить следующие действия:

1. Выделить группируемые значения из области заголовков сводной таблицы.
2. Выполнить команду *Данные → Группа и структура → Группировать*. В результате Excel создаст новое поле *Группа1*, в котором будут находиться сгруппированные данные (рис. 1.50).

	A	B	C	D	E	F	G
1							
2							
3	Сумма по полю	Сумма	Год				
4	Город2	Город	1993	1994	1995	1996	Общий итог
5	Гомель	Гомель	5000	6000	7000	8000	26000
6	Группа1	Мозырь	4000	3000	2000	4000	13000
7		Речица	2000	3000	3000	4000	12000
8		Гомель Сумма	5000	6000	7000	8000	26000
9		Мозырь Сумма	4000	3000	2000	4000	13000
10		Речица Сумма	2000	3000	3000	4000	12000
11	Общий итог		11000	12000	12000	16000	51000

Рис. 1.50. Пример сгруппированных данных сводной таблицы

Для удаления группировки следует выбрать команду *Данные → Группа и структура → Разгруппировать*, выделив предварительно сгруппированную область.

Удаление сводной таблицы

Для удаления сводной таблицы необходимо выбрать какую-либо ее ячейку. Затем на панели инструментов *Сводные таблицы* нажать кнопку *Сводная таблица* и в появившемся меню выбрать команду *Выделить → Таблицу целиком*, после чего выполнить команду *Правка → Очистить → Все*.

Создание диаграмм по сводной таблице

Поскольку сводная таблица – это диапазон на листе рабочей книги, то по ее данным можно построить диаграмму. Если все параметры будут установлены правильно, то диаграмма будет автоматически изменяться при изменении сводной таблицы. Однако при построении диаграммы следует придерживаться следующих правил:

- не отображать в сводной таблице промежуточные и общие итоги, поскольку это разъединит диапазон данных;
- не использовать более двух полей для элементов строки и столбца;
- перед созданием диаграммы полностью выделить сводную таблицу за исключением поля страницы.

1.13. Анализ данных с использованием надстроек Excel

1.13.1. Использование сценариев

Одной из наиболее важных сторон электронных таблиц является возможность использования формул для построения динамических моделей, которые немедленно реагируют на изменение исходных данных. Формулы автоматически пересчитывают результат, если изменяется содержимое ячеек, которые в них используются. Процесс изменения значений в ячейках и отслеживания результатов ячеек, в которых содержатся формулы, является одним из видов анализа данных, который называется «что – если».

Уникальное значение в ячейке или уникальная группа значений для диапазона ячеек называется *сценарием*. Сценарии позволяют проводить анализ «что – если». В ключевые ячейки можно вводить различные значения и смотреть, что при этом происходит. Сценарии предоставляют пользователю возможность обновлять различные варианты решения.

1.13.2. Использование таблиц подстановки

При работе с моделью «что – если» возникает необходимость сравнивать результаты нескольких сценариев. Для решения этой проблемы используется возможность Excel – *Таблицы подстановки*. Они позволяют автоматически создавать итоговые таблицы при различных комбинациях исходных данных. Однако эти таблицы имеют ограничение – они могут одновременно оперировать только с одной или двумя ячейками исходных данных. Таким образом, нельзя создать таблицу подстановки, которая использовала бы комбинацию трех или более ячеек с исходными данными.

Таблицу можно расположить в любом месте рабочей таблицы. Левый столбец содержит различные значения входного параметра. Верхняя строка содержит формулы или ссылки на ячейки с формулами, по которым рассчитывается результат. Причем сами формулы находятся в другом месте листа рабочей таблицы. Можно использовать любое количество ссылок на формулы. Верхняя левая ячейка таблицы не используется. Excel вычисляет значения, которые получаются в результате подстановки каждого из исходных значений во входную ячейку и помещает результат в соответствующий столбец (в ячейки, находящиеся под ячейками с соответствующими формулами или ссылками на формулы). Для создания таблиц подстановки существует команда *Данные → Таблица подстановки*.

Пример

Пусть дана таблица, представленная на рис. 1.51, которая обрабатывает информацию по ипотечной ссуде. Левая часть листа разделена на две секции: *Исходные данные* (ячейки для ввода) и *Результаты расчета* (ячейки результатов), которые вычисляются по формулам:

$\text{Размер ссуды} = \text{Цена} * (1 - \text{Первый взнос});$

$\text{Месячная плата} = \text{ППЛАТ}(\text{Процентная ставка}/12);$

$\text{Срок погашения}; - \text{Размер ссуды};$

$\text{Общая сумма} = \text{Месячная плата} * \text{Срок погашения};$

$\text{Общая сумма комиссионных} = \text{Общая сумма} - \text{Цена}.$

Функция ППЛАТ относится к категории финансовых функций и возвращает величину выплаты за один период годовой ренты.

Синтаксис: ППЛАТ(Норма; Кпер; Нз), где *Норма* – норма прибыли за период займа; *Кпер* – общее число периодов годовой ренты; *Нз* – текущая стоимость; общая сумма всех будущих платежей с настоящего момента.

Необходимо создать таблицу, в которой бы отражались значения, рассчитанные по формулам, находящимся в четырех ячейках (*Размер ссуды*, *Месячная плата*, *Общая сумма*, *Общая сумма комиссионных*), при изменении процентных ставок от 7% до 9% с шагом 0,25%.

	A	B	C	D	E	F	G	H	I
1	Расчет ипотечной ссуды				Таблица подстановки				
2					8%	161 520р.	1 185р.	426 664р.	224 764р.
3	Исходные данные				7,00%	161520	1074,597	386854,8	184954,8
4	Цена		201 900р.		7,25%	161520	1101,851	396666,4	194766,4
5	Первый взнос		20%		7,50%	161520	1129,371	406573,7	204673,7
6	Срок погашения ссуды		360		7,75%	161520	1157,149	416573,7	214673,7
7	Процентная ставка		8%		8,00%	161520	1185,177	426663,6	224763,6
8					8,25%	161520	1213,446	436840,5	234940,5
9	Результаты расчета				8,50%	161520	1241,949	447101,7	245201,7
10	Размер ссуды		161 520р.		8,75%	161520	1270,678	457444,3	255544,3
11	Месячная плата		1 185р.		9,00%	161520	1299,626	467865,5	265965,5
12	Общая сумма		426 664р.						
13	Общая сумма комиссионных		224 764р.						

Рис. 1.51. Расчетная таблица примера таблицы подстановки

Решение

Решение примера предполагает следующий порядок действий:

1. Создать шаблон для таблицы подстановки (см. правую часть листа на рис. 1.51), в которой строка 2 состоит из ссылок на соответствующие ячейки с формулами (например, ячейка F2 содержит формулу =C10, ячейка G2 – формулу =C11 и т. д.). Столбец E содержит варианты входных значений (процентных ставок), которые будут использоваться при расчетах.

2. Отметить диапазон ячеек E2:I11, в котором будет находиться таблица подстановки.

3. Выполнить команду *Данные → Таблица подстановки*. Появится одноименное диалоговое окно, приведенное на рис. 1.52, в котором необходимо определить ячейку листа, в которую должны подставляться исходные данные.

4. В поле *Подставлять значения по строкам* ввести ссылку на ячейку \$C\$7.

5. Нажать *ОК* и Excel поместит в ячейки таблицы подстановки значения формул массивов. При изменении ссылок на ячейки в первой строке таблицы или изменении значений процентных ставок в первом столбце, значения в созданной таблице будут автоматически пересчитываться.

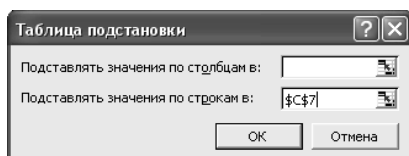


Рис. 1.52. Диалоговое окно *Таблица подстановки*

1.13.3. Диспетчер сценариев

Диспетчер сценариев позволяет автоматизировать процесс выполнения анализа «что – если» для различных моделей. В отличие от таблиц подстановки, которые одновременно анализируют данные только при изменении одного или двух исходных параметров, диспетчер сценариев позволяет создать несколько наборов данных вводимых значений для любого количества переменных и присвоить имя каждому набору. Эти значения называются *изменяемыми ячейками*. По имени можно выбрать определенный набор данных, и Excel покажет результаты анализа этих данных на листе рабочей таблицы. Кроме того, можно создать итоговый отчет по сценариям, в котором будет показан результат подстановки различных комбинаций входных параметров. Отчет может быть представлен в виде обычного структурированного списка или сводной таблицы.

Для вызова диспетчера сценариев используется команда *Сервис → Сценарии*. При нажатии кнопки *Добавить* появляется диалоговое окно *Добавление сценария* в диспетчер, приведенное на рис. 1.53, которое содержит ряд элементов управления:

- *Название сценария*, где указывается имя сценария.
- *Изменяемые ячейки* – ячейки, в которых находятся исходные данные для сценария. Позволяет ввести абсолютный адрес ячейки или ее имя. Можно выбирать несколько ячеек, не обязательно смежных. Количество изменяемых ячеек для одного сценария ограничено числом 32.
- *Примечание* – по умолчанию в это поле Excel помещает информацию о том, кто создал сценарий, а также дату его создания.
- *Защита* позволяет защитить сценарий от изменений и скрыть его.

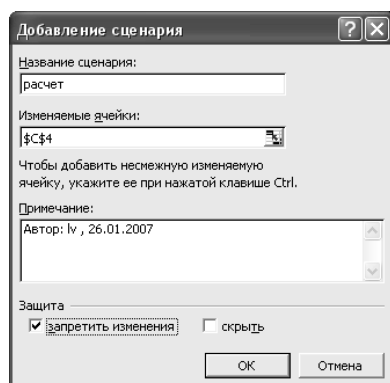


Рис. 1.53. Диалоговое окно *Добавление сценариев*

1.13.4. Процедура «Подбор параметра»

Данная процедура позволяет выполнить анализ, обратный сценарию «что – если», т. е. найти исходное данное, которое, будучи подставленным в формулы, даст необходимое значение в ячейку результата. *Подбор параметра* определяет значение одной входной ячейки, которое требуется для получения желаемого результата в зависимой ячейке (ячейке результата). Другими словами, эта процедура находит значение только для одной изменяемой ячейки и выдает только одно решение. Подбор параметра является частным случаем процедуры поиска решения. Он реализуется следующей последовательностью действий:

1. Создать сценарий «что – если».
2. Активизировать целевую ячейку (результата сценария).
3. Выполнить команду *Сервис → Подбор параметра*.
4. В одноименном диалоговом окне, приведенном на рис. 1.54, ввести значение, которое должно рассчитаться в целевой ячейке.

	A	B	C	D	E	F	G
1	Расчет ипотечной ссуды						
2							
3	Исходные данные						
4	Цена		201 900р.				
5	Первый взнос		20%				
6	Срок погашения ссуды		360				
7	Процентная ставка		8%				
8							
9	Результаты расчета						
10	Размер ссуды		161 520р.				
11	Месячная плата		1 185р.				
12	Общая сумма		426 664р.				
13	Общая сумма комиссионных		224 764р.				

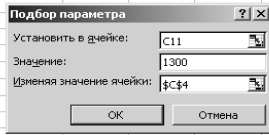


Рис. 1.54. Расчетная таблица примера на подбор параметра

5. Указать исходную ячейку сценария, которую можно изменять.

6. Нажать **ОК**.

7. При корректно введенных данных Excel объявит, что решение найдено и выведет окно *Результат подбора параметра*. В этом окне будет отображено подбираемое значение и значение, предложенное Excel. В нем же есть две кнопки:

- **ОК** для замены прежнего значения найденным.
- **Отмена** для возврата таблицы в прежнее состояние.

Пример

На основе таблицы, приведенной на рис. 1.51, необходимо рассчитать размер ссуды, которую можно взять (C11), если в месяц есть возможность платить 1300 р. для ее погашения (\$C\$4).

Решение

Пример решается обеспечением настройки диалогового окна *Подбор параметра*, приведенного на рис. 1.54. В результате подбора параметра (*Сервис* → *Подбор параметра*) Excel находит решение: при месячной плате 1 300 р. можно взять ссуду на 221 461 р.

1.13.5. Процедура «Поиск решения»

Данная процедура позволяет найти значения в нескольких входных ячейках, которые требуются для получения желаемого результата. Более того, накладывая некоторые ограничения на множество данных, можно получить одно или несколько решений задачи с большой долей вероятности.

Процедура *Поиск решения* позволяет находить решение, при котором значение в определенной ячейке листа рабочей таблицы достигает максимального, минимального или конкретного значения. Задачи при этом должны удовлетворять следующим критериям:

- значение в *целевой ячейке* должно зависеть от других ячеек и формул. Необходимо определить все исходные параметры, при которых значение в целевой ячейке будет максимальным, минимальным или заранее определенным;
- целевая ячейка зависит от группы ячеек, которые называются *изменяемыми ячейками*. Их значения должны быть подобраны таким образом, чтобы повлиять на результат в целевой ячейке;
- решение должно находиться в определенных пределах или удовлетворять определенным *ограничениям*.

Таким образом, *Поиск решения* позволяет решать задачи линейного и нелинейного программирования и находить оптимальные решения. Процедура реализуется командой *Сервис* → *Поиск решения*. На экране появляется диалоговое окно, приведенное на рис. 1.55. Оно содержит следующие элементы управления:

- в поле *Установить целевую ячейку* приводится ссылка на ячейку с функцией, значение которой (максимальное или минимальное) *Поиск решения* будет искать, изменяя значения параметров таким образом, чтобы они удовлетворяли налагаемым на них ограничениям;
- группа *Равной* устанавливает тип взаимосвязи между решением и целевой ячейкой (максимальное, минимальное или конкретное значение);
- в поле *Изменяя ячейки* приводится ссылка на диапазон ячеек или группу диапазонов ячеек, отведенных под неизвестные. Значения в этих ячейках должны изменяться в процессе поиска решения задачи для того, чтобы найти решение, удовлетворяющее заданным ограничениям;
- список *Ограничения* позволяет вводить ограничения в виде равенств, неравенств и целых значений.

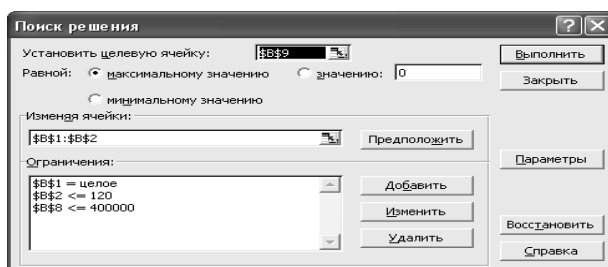


Рис. 1.55. Диалоговое окно Поиск решения

Для *Поиска решения* можно также устанавливать параметры ограничения времени решения задачи, предельное число итераций для ограничения промежуточных вычислений, относительную погрешность и метод поиска в виде алгоритма оптимизации.

1.13.6. Анализ данных с помощью инструментов «Пакета анализа»

Пакет анализа – это надстройка, обеспечивающая доступ к средствам анализа, которые не входят в стандартную установку Excel. Вызывается командой *Сервис* → *Анализ данных* (если надстройка отсутствует, необходимо выбрать команду *Сервис* → *Надстройки* → *Пакет анализа*).

Пакет анализа состоит из *двух частей*:

1. Аналитические процедуры:

- дисперсионный анализ (три разновидности);
- корреляционный анализ;
- описательная статистика;
- экспоненциальное сглаживание;
- двухвыборочный F-тест для дисперсии;
- анализ Фурье;
- гистограмма;
- скользящее среднее;
- генерация случайных чисел;
- ранг и перцентиль;
- регрессия;
- выборка;
- t-тест (три вида);
- z-тест.

2. Встроенные функции листа рабочей таблицы – это функции категорий (дата и время, инженерные, финансовые, информационные, математика и тригонометрия), доступные посредством *Мастера функций*.

Рассмотрим использование некоторых инструментов *Пакета анализа*.

Дисперсионный анализ

Дисперсионный анализ – это статистический тест, который определяет, взяты ли выборки из одной и той же совокупности. Пакет анализа позволяет выполнять дисперсионный анализ трех видов:

1. Однофакторный дисперсионный анализ – простой дисперсионный анализ с использованием только одной выборки для каждой группы данных.

2. Двухфакторный анализ с повторениями – расширенный дисперсионный анализ с несколькими выборками (или повторениями) для каждой группы данных.

3. Двухфакторный анализ без повторений – расширенный дисперсионный анализ с одной выборкой (повторением) для каждой группы данных.

Корреляционный анализ

Метод корреляции используется для оценки степени различия двух наборов данных. Степень корреляции выражается коэффициентом, который лежит в интервале от $-1,0$ (полностью отрицательная корреляция) до $+1,0$ (полностью положительная корреляция). Коэффициент 0 означает, что два набора не связаны.

Например, если наивысшие значения в одном наборе данных связаны с наивысшими значениями во втором наборе данных, то у этих наборов положительная корреляция.

Ковариационный анализ

При использовании средств ковариационного анализа создается матрица, аналогичная корреляционной матрице. Ковариация определяется как среднее от произведения отклонений каждой пары точек данных от их относительных средних. Она измеряет степень различия двух переменных аналогично корреляции.

Экспоненциальное сглаживание

Экспоненциальное сглаживание – это метод прогнозирования данных на основе предыдущего значения выборки и предыдущего предсказанного значения. Позволяет определить *фактор затухания* (называемый *константой сглаживания*), который выбирается из интервала от 0 до 1. Этот фактор определяет, насколько значимы предыдущая выборка и предсказанное значение. Выводит стандартные погрешности и позволяет строить диаграммы.

Генерация случайных чисел

В Excel предусмотрена специальная встроенная функция для вычисления случайных чисел, однако ин-

струмент *Пакета анализа*, генерирующий случайные числа, является более удобным, так как позволяет определять тип распределения для случайных чисел:

- *Равномерное* (генерируется последовательность случайных чисел в заданном интервале с одинаковой вероятностью).
- *Нормальное* (для чисел задается среднее значение и стандартное отклонение).
- *Бернулли* (случайные числа имеют значение 0 или 1 в зависимости от заданной вероятности успеха).
- *Дискретное* (позволяет определить вероятность, характеризующую выбираемые значения. Для нее требуется входной диапазон, состоящий из двух столбцов: в первом содержатся значения, во втором – вероятности выбора каждого значения. Сумма вероятностей во втором столбце должна равняться 1).

1.14. Базовые сведения о VBA

1.14.1. Возможности VBA

VBA (Visual Basic for Applications) – это язык программирования, который позволяет конструировать профессиональные приложения и решать практические задачи в среде Windows. VBA применяет технологию визуального программирования, т. е. конструирование рабочей поверхности приложения и элементов его управления непосредственно на экране, а также запись всей программы или ее частей с помощью макросов.

Макрос – это программа, которая автоматизирует некоторые функции Excel, позволяя выполнять работу в приложении более эффективно. С помощью макросов можно производить следующие действия:

- автоматизировать часто выполняемую процедуру;
- автоматизировать повторяющиеся операции;
- создать новую команду;
- создать новую кнопку панели инструментов;
- создать упрощенный внешний интерфейс пользователя;
- разработать новую функцию рабочей таблицы;
- создать законченные управляемые приложения;
- создать новые надстройки для Excel.

Макросы могут быть *двух видов*:

1. *VBA-процедура* – новая команда, которая может быть выполнена пользователем либо другим макросом. На листе модуля может содержаться произвольное число процедур. Процедура всегда начинается с ключевого слова *Sub*, после которого следует имя макроса (у каждого макроса должно быть уникальное имя), а затем круглые скобки. Эти скобки обязательны, но если у процедуры нет аргументов, они остаются пустыми. Заканчивается процедура оператором *End Sub*. Строки, заключенные между операторами *Sub* и *End Sub*, называются *телом процедуры* или *текстом макроса*.

2. *VBA-функция* – запускается из VBA-процедуры и используется в формулах рабочей таблицы точно так же, как и встроенные функции рабочих таблиц Excel. Функция начинается ключевым словом *Function* и заканчивается оператором *End Function*.

1.14.2. Понятие об объектной модели

VBA относится к языкам объектно ориентированного программирования. Другими словами, к нему применима методика анализа, проектирования и написания приложений с применением объектов. Рассмотрим основные понятия объектно ориентированного программирования.

Объект – это совокупность данных вместе с кодом, предназначенным для их обработки. Объекты располагаются в иерархическом порядке и могут содержать подчиненные объекты. Основными объектами языка VBA являются следующие: *Application* – главный (корневой) объект в иерархии объектов Excel, представляющий само приложение; *Workbook* – файл рабочей книги; *Worksheet* – рабочий лист; *Range* – диапазон; *Chart* – диаграмма; *Window* – окно приложения Excel.

Семейство объектов представляет собой объект, содержащий несколько других объектов одного типа. Каждый элемент семейства нумеруется и может быть идентифицирован либо по номеру, либо по имени (например, существуют семейства *Workbooks*, *Worksheets*, *Ranges*, *Charts*, *Windows*).

Класс – это проект, на основе которого впоследствии будет создан конкретный объект. Таким образом, класс определяет имя объекта, его свойства и действия, выполняемые над объектом. В свою очередь каждый объект является *экземпляром класса*.

Свойство представляет собой атрибут объекта, определяющий его характеристики, такие, как размер, цвет, положение на экране и состояние объекта (например, доступность или видимость). Для изменения характеристики объекта необходимо изменить значения его свойств. Свойство можно изменять сразу у всех объектов семейства. *Синтаксисы* установки и чтения значения свойства объекта выглядят соответственно следующим образом:

Объект.Свойство = Выражение;
Переменная = Объект.Свойство

Метод – это действие, выполненное над объектом. Метод можно применять ко всем объектам семейства. *Синтаксис*: Объект.Метод (аргументы).

Аргументы у методов могут отсутствовать. Между собой аргументы разделяются пробелами. Таким образом, можно сказать, что *объект* – это элемент программы, который имеет свое отражение на экране, содержит некоторые переменные, представляющие его свойства, и некоторые методы для управления самим объектом.

Событие представляет собой действие, распознаваемое объектом, для которого можно запрограммировать отклик. События возникают в результате действий пользователя или программы, или могут быть вызваны системой.

1.14.3. Основные принципы программирования на VBA

Суть программирования на VBA заключается в двух понятиях: *событие* и *отклик* на него. Если пользователь производит какое-то воздействие на систему, то в качестве отклика выполняется код созданной пользователем процедуры. Если такой отклик не создан, т. е. не написана соответствующая процедура, то система не реагирует на данное событие, и оно остается безответным. Таким образом, действия, происходящие в системе, являются событиями, а отклики на них – процедурами. Этот специальный вид процедур, генерирующий отклик на событие, называется *процедурами обработки событий*. Таким образом, программирование на VBA состоит в создании кода программ, которые генерируют отклики на события.

Рассмотрим основные *принципы* работы с VBA.

Необходимые действия записываются в виде команд на листе VBA-модуля, после чего макрос запускается на выполнение. VBA-модули сохраняются в рабочей книге Excel. Их может быть любое количество.

VBA включает все конструкции современного языка программирования, включая массивы, циклы и т. д.

VBA-модуль состоит из *процедур-подпрограмм*. Подпрограмма – это компьютерная программа, которая выполняет некоторые действия над объектом.

VBA-модуль может содержать *процедуры-функции*. Особенность функции состоит в том, что она возвращает единственное значение. Функцию можно вызвать из другой VBA-процедуры, а также использовать в формуле рабочей таблицы.

Основным элементом VBA является *объект*. Excel содержит множество объектов, которыми можно оперировать (к ним относятся рабочая книга, рабочая таблица, диапазон ячеек, диаграмма и др.).

Объекты имеют свою иерархию. В основе иерархического принципа лежит способность одних объектов служить контейнерами для других (или содержать другие объекты).

Например, приложение Excel (объект Application) может содержать другие объекты-рабочие книги (Workbook), панели инструментов (Toolbar) и др. Объект Workbook может содержать такие объекты, как листы рабочих таблиц (Worksheet), листы диаграмм (Chart) и др.

Подобные объекты составляют *семейства* или *коллекции*. Например, семейство Worksheets содержит все листы рабочих таблиц некоторой рабочей книги. Семейство Toolbars содержит все панели инструментов. При этом сами наборы также являются объектами.

Для обращения к объекту необходимо указать его положение в иерархии объектов, используя точку в качестве разделителя. Например, для того, чтобы обратиться к рабочей книге под именем Book1.xls, необходимо записать: Application.Workbooks("Book1").

Данная запись вызывает обращение к рабочей книге Book1, входящей в семейство Workbooks, которая, в свою очередь, содержится в объекте Application (т. е. Excel). Если необходимо дополнительно указать Лист1 рабочей книги, то следует записать: Application.Workbooks("Book1").Worksheets("Sheet1").

У объектов есть свои *свойства*, которые определяют параметры объекта. С помощью VBA можно не только определить свойства объекта, но и изменить их. Например, объект Range (диапазон) имеет свойства Value (значение), Name (имя) и другие; объект Chart (диаграмма) имеет свойства HasTitle (заголовок), Type (тип).

При обращении к свойству, сначала необходимо обратиться к объекту, а затем к самому свойству, используя точку в качестве разделителя. Например, для того, чтобы обратиться к значению, находящемуся в ячейке A1 на листе Лист1, необходимо записать: Worksheets("Лист1").Range("A1").Value.

Переменным можно присваивать значения. Например, для присвоения переменной Interest значения из ячейки Лист1!A1 необходимо воспользоваться оператором VBA: Interest = Worksheets("Лист1").Range("A1").Value.

У объектов есть *методы*, т. е. действия, выполняемые над объектом. Например, одним из методов для объекта Range (диапазон) является ClearContents (очистить содержимое). Для указания диапазона, необходимо сначала обратиться к объекту, а затем – к методу, используя точку в качестве разделителя. Например, для того, чтобы очистить содержимое ячейки A1, необходимо записать: Range("A1").ClearContents.

1.15. Интегрированная среда разработки

1.15.1. Работа с макрорекордером

Макрорекордер – это транслятор, создающий программу (макрос) на языке VBA, которая является результатом перевода действий пользователя на язык VBA с момента запуска макрорекодера до окончания записи макроса.

Для записи макроса необходимо выполнить следующие действия:

1. Активизировать макрорекордер командой *Сервис → Макрос → Начать макрос*.
2. Появится диалоговое окно *Запись макроса*, приведенное на рис. 1.56, которое позволяет задать параметры макроса, назначить макросу комбинацию клавиш, выбрать рабочую книгу для сохранения макроса.
3. К параметрам макроса относятся *Имя макроса* и *Описание*. В диалоговом окне им соответствуют одноименные поля. По умолчанию макросам присваиваются имена Макрос1, Макрос2, ..., однако удобно не использовать эти стандартные имена, а присваивать макросу уникальное имя. *Описание* используют для многократно выполняемых макросов в виде подсказки.

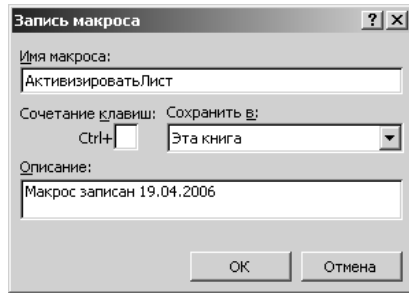


Рис. 1.56. Диалоговое окно *Запись макроса*

4. Для назначения макросу комбинации клавиш используют поле *Сочетание клавиш*. В нем следует указать символ, который в комбинации с клавишей *Ctrl* позволит выполнить макрос. Однако этот параметр необязательный и применяется только для часто используемых макросов и быстрого доступа к ним. Не используя комбинацию клавиш, макрос можно вызвать командой *Сервис → Макрос → Макросы*.

5. Для сохранения макроса используют раскрывающийся список *Сохранить* данного диалогового окна, который имеет *три* опции:

- *Личная книга макросов* сохраняет макрос в специальную скрытую книгу, в которой хранятся макросы. Книга всегда открыта, хотя и скрыта, а записанные в ней макросы доступны для других рабочих книг. Личную книгу можно отобразить по команде *Окно → Отобразить*.
- *Эта книга* по умолчанию предлагается компьютером. Макрос сохранится на новом листе модуля в активной рабочей книге.
- *Новая книга* – макрос сохранится в новой рабочей книге.

6. Нажать *ОК*. Появится плавающая панель инструментов с кнопкой *Остановить запись*. Все производимые действия будут записываться до тех пор, пока кнопка не будет нажата или не будет вызвана команда *Сервис → Макрос → Остановить запись*.

7. Для просмотра записанной процедуры используется команда *Сервис → Макрос → Макросы*. Она вызывает диалоговое окно *Макрос*, в котором представлен список макросов. При выборе кнопки *Изменить* на экране отобразится окно редактора VBA с активизированным стандартным модулем, в котором будет находиться код (листинг) записанного макроса.

1.15.2. Редактор VBA

Код VBA набирается в редакторе Visual Basic. Вызов редактора осуществляется по команде *Сервис → Макрос → РедакторVisualBasic*. В результате открывается интегрированная среда разработки приложений редактора Visual Basic, приведенная на рис. 1.57. Для возврата в рабочую книгу используют кнопку *View Microsoft Excel* панели инструментов *Стандартная*.

Интегрированная среда VBA включает в себя следующие *компоненты*:

- окно проекта;
- окно редактирования кода;
- окно форм;
- окно свойств;
- окно просмотра объектов.

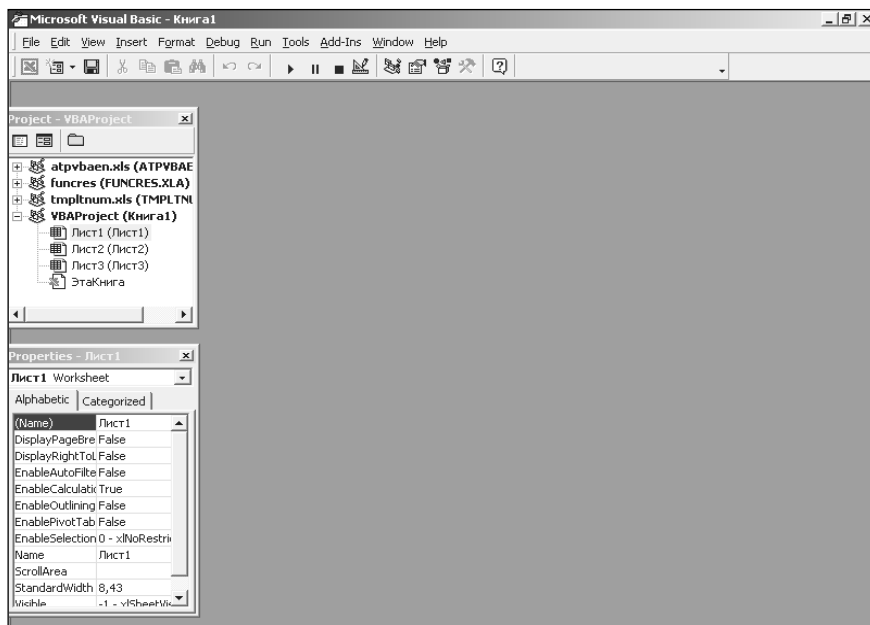


Рис. 1.57. Интегрированная среда VBA

1.15.3. Окно проекта

В окне проекта, пример которого приведен рис. 1.58, выводятся проекты всех открытых рабочих книг. Окно активизируется командой *View* → *Project Explorer*. В окне представлена иерархическая структура модулей текущего проекта (*модуль* – это текстовый файл, в котором набирается код). Модули создаются для каждой пользовательской формы, макроса и класса.

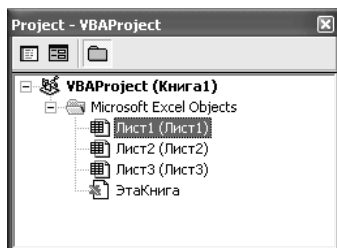


Рис. 1.58. Пример окна проекта

По своему назначению модули делятся на два *типа*:

- *модули объектов* – это модули, связанные с рабочей книгой, рабочими листами, формами, а также модули класса. Формы создаются выбором команды *Insert* → *UserForm*, а модули класса – *Insert* → *ClassModule*;
- *стандартные* – это модули, на которых записываются макросы. Такие модули добавляются в проект командой *Insert* → *Module*.

1.15.4. Окно редактирования кода

Окно редактирования кода, пример которого приведен рис. 1.59, служит для ввода кода процедур приложения. Код внутри модуля организован в отдельные разделы для каждого объекта, программируемого в модуле.

Открыть модуль в редакторе кода для соответствующего объекта можно командой *View* → *Code* или двойным щелчком по объекту в окне проекта. Вернуться из модуля к объекту можно командой *View* → *Object*.

В окне доступны два режима представления кода: просмотр отдельной процедуры и просмотр всего модуля. Переключение режимов осуществляется выбором кнопок *Procedure View* или *Full Module View* внизу окна редактирования.



Рис. 1.59. Пример окна редактирования кода

Для ориентации в процедурах в первой строке окна используются два раскрывающихся списка: *General* (позволяет выбрать объект) и *Declarations* (содержит список событий, допустимых для выбранного в предыдущем списке объекта).

1.15.5. Окно форм

Формы используются для создания диалоговых окон разрабатываемых приложений в VBA. Редактор форм является одним из основных инструментов визуального программирования. Форма добавляется в проект командой *Insert → UserForm*. В результате на экран выводится незаполненная форма с панелью инструментов *Toolbox*, как показано на рис. 1.60. Используя эту панель, можно сконструировать любое требуемое диалоговое окно для приложения.

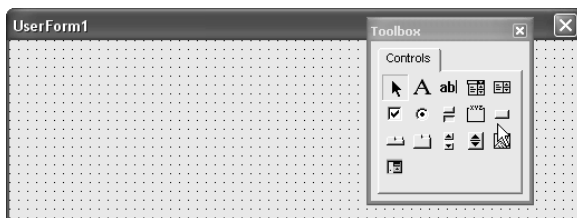


Рис. 1.60. Незаполненная форма с панелью инструментов *Toolbox*

1.15.6. Окно свойств

В этом окне, пример которого приведен рис. 1.61, перечисляются основные установки свойств выбранной формы или элемента управления. Для просмотра свойств выбранного объекта используют команду *View → Properties Windows*.

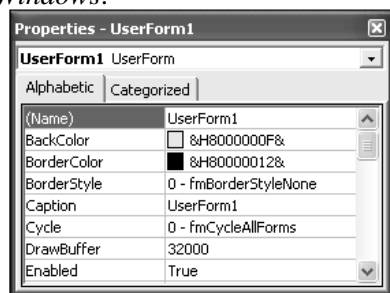


Рис. 1.61. Пример окна свойств

Окно свойств состоит из двух составных частей: *верхней* и *рабочей*. В *верхней части* окна располагается раскрывающийся список, из которого можно выбрать любой элемент управления текущей формы или саму форму. *Рабочая часть* окна состоит из двух вкладок: *Alphabetic* и *Categorized*, которые отображают набор свойств в алфавитном порядке или по категориям. Для изменения значений свойств используют ввод с клавиатуры соответствующего значения свойства в поле или выбирают его из раскрывающегося списка данного поля.

1.15.7. Окно просмотра объектов

Это окно, пример которого приведен рис. 1.62, отображает список всех объектов, которые имеются в системе и которые можно использовать при создании проекта. Оно активизируется командой *View → Object Browser*.

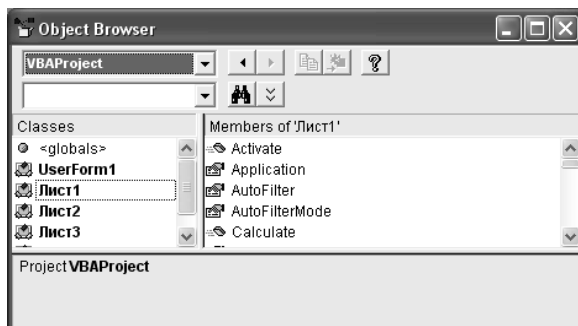


Рис. 1.62. Пример окна просмотра объектов

Окно состоит из трех частей:

- раскрывающийся список *Project Library (All Library)*, из которого можно выбрать различные проекты и библиотеки объектов. Например, проекты, библиотеки объектов Excel, VBA Office и др.);
- список *Classes* позволяет просматривать все классы объектов выбранной библиотеки;
- список *Members* позволяет просматривать все компоненты выбранного класса.

1.16. Отладка программ

1.16.1. Типы ошибок при отладке программ

При написании программ пользователь может допускать те или иные ошибки. Условно ошибки можно разделить на три типа:

- ошибки компиляции;
- ошибки выполнения;
- логические ошибки.

1.16.2. Ошибки компиляции

Эти ошибки возникают в том случае, если Visual Basic не может интерпретировать введенный код. Например, при некорректном вводе числа скобок, неправильном имени, неполном вводе инструкции и т. д. Некоторые из этих ошибок обнаруживаются Visual Basic при завершении набора строки с инструкцией в редакторе кода нажатием клавиши *Enter*. Строка, в которой содержится ошибка, выделяется красным цветом, и на экране отображается диалоговое окно с сообщением о возможной причине, вызвавшей ошибку.

Другие ошибки компиляции обнаруживаются перед исполнением программы. VBA каждый раз автоматически компилирует программу при ее запуске на выполнение. В этом случае предполагаемое место-положение ошибки выделяется синим цветом, и на экране отображается диалоговое окно Visual Basic с сообщением о возможной причине, вызвавшей ошибку. Например, опущена инструкция *End Select* в операторе *Select*.

1.16.3. Ошибки выполнения

Эти ошибки возникают после успешной компиляции программы при ее выполнении. Причинами таких ошибок могут быть следующие:

- некорректная информация при считывании файла с диска;
- некорректные данные, введенные пользователем (например, требуется число, а пользователь вводит строковую информацию);
- некорректность вычислений (например, деление на ноль).

В этом случае на экране отображается диалоговое окно Visual Basic, приведенное на рис. 1.63, с сообщением о номере ошибки и возможной причине, ее вызвавшей. Если в диалоговом окне Visual Basic нажать кнопку *Debug*, то в строке модуля желтым цветом будет выделена строка, вызвавшая ошибку, на которой выполнение программы было прервано. Кроме того, эта строка будет помечена стрелкой и VBA перейдет в режим прерывания.

Одним из удобств режима прерывания является возможность узнать текущее значение переменных и свойств. Для этого достаточно расположить указатель мыши на имени свойства или переменной. Появится всплывающая подсказка с текущим значением переменной или свойства. Для задания режима вывода всплывающей подсказки необходимо вызвать команду *Tools → Options* и в диалоговом окне *Options* на вкладке *Editor* установить флажок *AutoDataTips*.

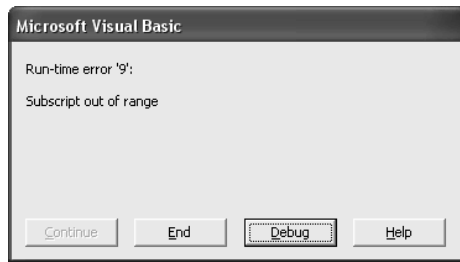


Рис. 1.63. Окно сообщения об ошибке выполнения

Кроме режима прерывания, приложение может находиться в *режиме разработки* и *режиме выполнения*. В режиме разработки непосредственно создается приложение – конструируются формы и набирается код. В режиме выполнения приложение получает управление для взаимодействия с пользователем.

Переключение между режимами производится при помощи *трех кнопок* панели инструментов *Standard*:

- *Start* доступна в режиме конструирования. Переключает в режим выполнения. В режиме прерывания она также доступна, но играет роль кнопки *Continue*.
- *Break* доступна в режиме выполнения. Переключает в режим прерывания.
- *End* доступна в режиме выполнения. Переключает в режим разработки.

1.16.4. Логические ошибки

Эти ошибки труднее всего обнаружить и устранить. Они не приводят к прерыванию выполнения программы, т. е. визуально все выглядит так, как будто программа работает правильно, однако результаты выдает неверные. Локализация логических ошибок связана с тщательным анализом алгоритма программы с привлечением средств отладки VBA.

1.16.5. Пошаговое выполнение программ

Редактор Visual Basic позволяет осуществлять пошаговое выполнение программы. Такой режим можно задать при помощи панели инструментов *Debug* или из меню *Debug*, которое включает команды и соответствующие комбинации клавиш.

Для выполнения программы в пошаговом режиме используются *четыре команды*:

- команда *Debug* → *Step Into* осуществляет последовательную, шаг за шагом, отладку всей программы, включая процедуры, вызываемые в программе;
- команда *Debug* → *Step Over* осуществляет последовательную, шаг за шагом, отладку всей программы, но не заходя в процедуры, вызываемые в программе. Если встречается процедура, то она выполняется целиком, а не шаг за шагом, как это делается в команде *Debug* → *Step Into*;
- команда *Debug* → *Step Out* завершает выполнение текущей процедуры и останавливается на следующей инструкции программы, откуда процедура была вызвана;
- команда *Debug* → *Run to Cursor* выполняет программу до инструкции, помеченной курсором.

1.16.6. Точки прерывания

VBA приостанавливает выполнение программы перед строкой кода, содержащей точку прерывания, и переключается в режим прерывания. Точка прерывания устанавливается или снимается командой *Debug* → *Toggle Breakpoint*. В модуле точки прерывания выделяются полосой кирпичного цвета и кругом того же цвета, как показано на рис. 1.64.

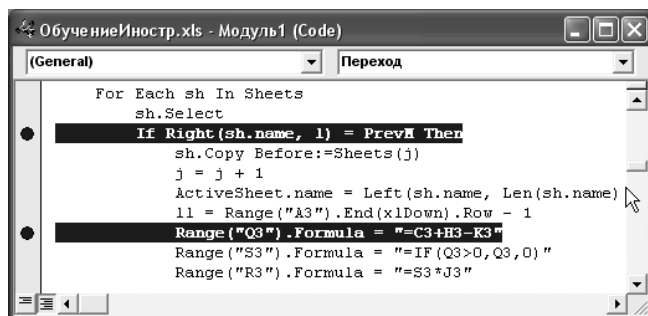


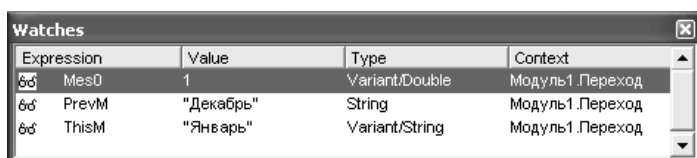
Рис. 1.64. Пример установленных точек прерывания

В одном проекте может быть несколько точек прерывания. Все инструкции, расположенные выше, между и ниже точек прерывания, выполняются в обычном режиме. Одновременно снять все точки прерывания можно командой *Debug* → *Clear All Breakpoint*.

1.16.7. Вывод значений свойств и переменных

Одним из удобств режима отладки является возможность узнать текущее значение переменных и свойств. Для этого используются специальные диалоговые окна.

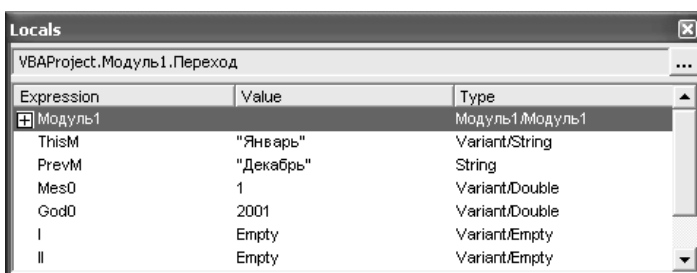
Окно *Watches* отображается на экране командой *View* → *Watch Window* или командой *Debug* → *Quick Watch*. Его пример приведен на рис. 1.65. Окно позволяет одновременно отображать текущие значения нескольких переменных или свойств. Команда *Debug* → *Add Watch* добавляет новые контрольные значения в диалоговое окно *Watches*. Удаление контрольного значения из диалогового окна производится его выделением и нажатием клавиши *Delete*.



Expression	Value	Type	Context
Mes0	1	Variant/Double	Модуль1.Переход
PrevM	"Декабрь"	String	Модуль1.Переход
ThisM	"Январь"	Variant/String	Модуль1.Переход

Рис. 1.65. Пример окна *Watches*

Окно *Locals* отображается на экране командой *View* → *Locals Window*. Оно выводит значения всех переменных текущей процедуры, а не только специально выбранных, как это происходит в окне *Watches*. Пример данного окна приведен на рис. 1.66.



Expression	Value	Type
Модуль1	Модуль1.Модуль1	
ThisM	"Январь"	Variant/String
PrevM	"Декабрь"	String
Mes0	1	Variant/Double
God0	2001	Variant/Double
I	Empty	Variant/Empty
II	Empty	Variant/Empty

Рис. 1.66. Пример окна *Locals*

Окно *Immediate* отображается на экране командой *View* → *Immediate Window*. Его пример приведен на рис. 1.67. Оно предоставляет пользователю следующие возможности:

- набирать и вычислять отдельные инструкции VBA. Для этого достаточно ввести в окне соответствующую инструкцию и нажать клавишу *Enter*. Единственным ограничением на инструкцию является то, что она должна быть набрана в одну строку;
- определять текущие значения переменных и свойств. Для этого в окне необходимо набрать вопросительный знак, имя переменной или свойства и нажать клавишу *Enter*;
- устанавливать новые текущие значения переменных. Для этого в окне необходимо набрать имя переменной, знак «=» и новое значение переменной;
- определять значения встроенных в VBA констант. Для этого в окне необходимо набрать вопросительный знак, имя константы и нажать клавишу *Enter*.

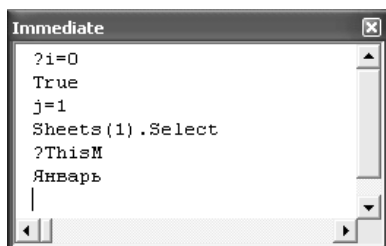


Рис. 1.67. Пример окна *Immediate*

1.17. Основы программирования на VBA

1.17.1. Типы данных и переменные

Типы данных относятся к самым фундаментальным понятиям любого языка программирования. Тип данных определяет множество допустимых значений, которое может принимать указанная переменная. В VBA существуют следующие типы данных: числовые, логический, строковый, дата и время, объектный, варьируемый.

К *числовым типам* относятся следующие подтипы:

- *Integer* (целое число) – размер 2 байта, диапазон значений от – 32 768 до 32 767.
- *Long* (длинное целое число) – размер 4 байта, диапазон значений от –2 147 483 648 до 2 147 483 647.
- *Single* (число с плавающей запятой одинарной точности) размер 4 байта, 6 разрядов после запятой, порядок: от E–45 до E38.
- *Double* (число с плавающей запятой двойной точности) – размер 8 байтов, 14 разрядов после запятой, порядок: от E–324 до E308.
- *Currency* (денежный) – размер 8 байтов, 4 разряда после запятой, диапазон значений от –9E14 до 9E14.

К числовым типам применяются *арифметические операции* (+, –, *, /, ^, Mod) и *операции сравнения* (<, >, <=, >=, =, <>).

Логический тип (Boolean) может принимать только два значения: *True* (истина) или *False* (ложь). К переменным этого типа применяются *логические операции*: And, Or, Not, Xor, Eqv и др.

Строковый тип (String) представляет собой массив постоянной или переменной длины, содержащий символы. Для того, чтобы определить строку постоянной длины, необходимо после ключевого слова *String* указать: *длина_строки. К строковым данным применимы *операции сравнения* (=) и *сцепления* (&), а также строковые функции.

Тип *Дата и время (Date)* позволяет хранить значения в диапазоне от 1 января 1900 г. 0:00:00 до 31 декабря 9999 г. 23:59:59. Для присвоения даты и времени переменным этого типа можно использовать выражения следующего вида:

- someDate = #31/3/1964#;
- someDate = #21:35#;
- someDate = #31/3/64 21:35#;
- someDate = #31 Март 1964#.

К переменным типа *Date* можно применять *арифметические операции* и *операции сравнения*, а также соответствующие функции.

Объектный тип (Object) позволяет в программах обращаться к свойствам объектов VBA. Как отмечалось выше, для управления объектами служат методы.

Варируемый тип (Variant) позволяет хранить данные любого из перечисленных выше типов и применяется по умолчанию к тем переменным, которые не были определены с помощью оператора *Dim*. Тип переменных *Variant* изменяется в зависимости от последнего присваивания.

1.17.2. Описание переменных

Переменная – это идентификатор (имя), определяющий область памяти для хранения величины, которая может изменяться во время работы программы. Описание типа каждой переменной делает программу надежнее и ускоряет ее работу, так как VBA не требуется тратить время на распознавание типа неописанной переменной при каждом обращении к ней.

Для описания *простых переменных* в языке VBA предназначен оператор *Dim*, имеющий *синтаксис*: Dim СписокПеременных As ТипПеременной:

- *Dim* – ключевое слово.
- *СписокПеременных* – список имен переменных, разделенных символом «запятая» и удовлетворяющих стандартным правилам именования переменных.
- *ТипПеременной* – тип данных переменной.

Примеры описания простых переменных:

- Dim Ставка As Single;
- Dim i, j, k As Integer;
- Dim myRange, someSheet As Object.

Для обязательного объявления всех переменных в начале модуля рекомендуется поместить директиву Option Explicit.

Ее использование не допускает возможность неправильного ввода имени переменной. Например, если переменная была объявлена как *Ставка*, а при наборе была использована латинская буква C, то это приведет к ошибке. В отсутствие директивы Option Explicit подобную ошибку трудно отследить.

1.17.3. Преобразование типов данных

Для преобразования типов данных существует набор функций, приведенных в табл. 1.5, с названиями, которые образуются путем добавления сокращенного имени типа к латинской букве C.

Таблица 1.5. Функции преобразования типов

Функция	Приведение к типу	Тип
CInt	Integer	Целый
CBool	Boolean	Логический
CCur	Currency	Денежный
CDate	Date	Дата и время
CLng	Long	Длинное целое
CDbl	Double	С плавающей запятой двойной точности
CSng	Single	С плавающей запятой обычной точности
CStr	String	Строковый
Cvar	Variant	Числовые подтипы

1.17.4. Константы

Константы, в отличие от переменных, не могут изменять свои значения. Использование констант позволяет проще вносить изменения в программу, так как отпадает необходимость многократно исправлять значения по тексту программы. Достаточно ввести новое значение при определении константы.

Для описания констант предназначен оператор *Const*, имеющий *синтаксис* (здесь и далее в квадратных скобках находятся необязательные элементы синтаксиса): *Const ИмяКонстанты [As Тип] = Выражение*.

ИмяКонстанты – имя, удовлетворяющее правилам именования переменных.

Тип – необязательный элемент описания, указывающий на один из поддерживаемых типов данных, кроме *Date*, *String*, *Variant*.

Выражение – константа или константное выражение, которое включает любые арифметические или логические операции.

Примеры использования констант:

- *Const Pi As Single = 3,14;*
- *Const S As String = "Это строка";*
- *Const EndDate As Date = #31/05/06#.*

1.17.5. Работа с массивами

Массив (array) – это группа однотипных переменных, которые объединены общим именем. Их можно считать одной из разновидностей переменных. Отличие состоит в том, что в массивах можно хранить не одно, а множество значений. Чаще всего используются массивы размерностей 1 (векторы) и 2 (матрицы). Массивы больших размерностей встречаются крайне редко. Максимальная размерность массива в языке VBA может быть 60.

Доступ к элементу одномерного массива (вектора) осуществляется с помощью одного индекса, к элементу двумерного массива (матрице) – двух индексов, разделенных символом «запятая», к элементу *k*-мерного массива – *k*-индексов, разделенных запятыми. По умолчанию базовым (начальным) индексом является 0.

Синтаксис объявления массива в отличие от переменной требует указания размерности массива и границ изменения индексов. Оператор *Dim* для массива имеет *синтаксис*: *Dim ИмяМассива(Список Индексов) As ТипМассива*.

ИмяМассива – имя массива, удовлетворяющее правилам именования переменных.

СписокИндексов – для одномерного массива (вектора) – это целочисленная константа, указывающая количество элементов массива; для двумерного массива (матрицы) – две целочисленных константы, разделенных символом запятая, где первая константа – количество строк матрицы, а вторая – количество столбцов; для *k*-мерного массива – *k* целочисленных констант, разделенных символом «запятая», указывающих на количество элементов по каждому измерению.

ТипМассива – тип данных массива.

Примеры объявления массивов:

- *Dim A(15) As Integer*. Объявлен одномерный массив (вектор) из 15 целых чисел, причем по умолчанию начальный элемент массива будет A(0), а последний A(14). В данном случае 0 – базовый (начальный) индекс.
- *Dim B(1 To 15) As Integer*. Объявлен одномерный массив также из 15 целых чисел, но базовым индексом является 1, а последним 15. Из примера следует, что для изменения базового индекса используется ключевое слово *To*.
- *Dim B(8, 10) As Single*. Объявлен двумерный массив (матрица) действительных чисел обычной точ-

ности, состоящий из 8 строк и 10 столбцов. Начальный элемент – В(0, 0), а последний – В(7, 9).

1.17.6. Процедуры и функции

Понятие процедуры

Процедура является самостоятельной частью кода, которая имеет имя и может содержать аргументы, выполнять последовательность инструкций и изменять значения своих аргументов.

Синтаксис:

```
Sub ИмяПроцедуры([СписокАргументов])  
    [ИнструкцииОписания]  
    УправляющиеИнструкции  
    [ExitSub]  
    [УправляющиеИнструкции]  
End Sub
```

ИмяПроцедуры – имя процедуры, удовлетворяющее правилам именования переменных.

СписокАргументов – список переменных, представляющий формальные аргументы, которые передаются в процедуру при ее вызове. Аргументы разделяются запятой. Они могут отсутствовать, но круглые скобки при этом все равно обязательны.

ИнструкцииОписания – группа инструкций, предназначенных для описания переменных процедуры.

УправляющиеИнструкции – группа инструкций, выполняемых в процедуре. Обычно каждая инструкция записывается в отдельной строке. Однако можно на одной строке разместить несколько инструкций, разделив их символом «двоеточие». Если необходимо продолжить инструкцию на новой строке, то последним символом текущей строки должен быть символ «подчеркивание».

Exit Sub приводит к немедленному выходу из процедуры.

End Sub оператор окончания процедуры.

Список аргументов имеет вид: [ByVal/ByRef] имяПеременной As ТипПеременной.

ByVal указывает, что этот аргумент передается по значению. Вследствие этого значение данного аргумента при выходе из процедуры недоступно в вызывающей процедуре.

ByRef указывает, что этот аргумент передается по ссылке. Используется по умолчанию. Значение данного аргумента при выходе из процедуры доступно в вызывающей процедуре.

ИмяПеременной – имя, удовлетворяющее правилам именования переменных.

ТипПеременной – тип данных аргумента.

Процедура, не имеющая аргументов, может быть вызвана из интегрированной среды VBA, при возникновении некоторого события, из другой процедуры. В первых двух случаях имя процедуры указывается в соответствующих диалоговых окнах. В последнем случае внутри вызывающей процедуры должен встретиться *оператор вызова процедуры*, имеющий два варианта *синтаксиса*: Call ИмяПроцедуры или ИмяПроцедуры.

Процедура с аргументами вызывается только из другой процедуры. Первый способ вызова процедуры заключается в использовании оператора Call, имеющего *синтаксис*: Call ИмяПроцедуры(Список Аргументов).

СписокАргументов – список фактических аргументов, представленных константами, переменными или выражениями, которые передаются в процедуру при ее вызове. Аргументы разделяются запятой и должны соответствовать по порядку следования, количеству и типу списка формальных аргументов, заданных в описании вызываемой процедуры.

Для второго способа вызова процедуры достаточно указать имя вызываемой процедуры и список фактических аргументов, который не заключается в круглые скобки: ИмяПроцедуры СписокАргументов.

Для второго способа имеется частный случай (часто встречается при работе с макрорекордером). Фактические аргументы могут несоответствовать по порядку следования формальным аргументам вызываемой процедуры. В этом случае нужно указать имя формального аргумента из описания вызываемой процедуры, затем поставить двоеточие и знак равенства, после которого помещается значение фактического аргумента.

Пример

Необходимо показать отличие передачи параметров по значению и по ссылке, а также способы вызова процедуры с аргументами.

Решение

```
Sub Demo(ByVal a, b, ByRef c)  
' a передается по значению и принимает значение первого аргумента  
' по умолчанию b передается по ссылке, соответствуя 2-му аргументу  
' c передается по ссылке, соответствуя 3-му аргументу  
    a = a + 1  
    b = b + a  
    c = c + b
```



```

End Sub
Sub Test()
    x = 1 : y = 10 : z = 100
    Call Demo(x, y, z)
' x не изменится, так как передается по значению
' y станет равным 12, z равным 112
    Demo 2, x, y          ' x станет равным 4, y равным 16
    Demo b:=y, a:=0, c:=x ' y станет равным 17, x равным 21
End Sub

```

Понятие функций

Основное отличие функции от процедуры состоит в том, что она возвращает значение.

Синтаксис:

```

Function ИмяФункции([СписокАргументов]) [As Тип]
    [ИнструкцииОписания]
    УправляющиеИнструкции
    [ИмяФункции = Выражение]
    [Exit Function]
    [УправляющиеИнструкции]
    [ИмяФункции = Выражение]
End Function

```

Элементы процедуры *Sub* и функции *Function* аналогичны. Функция *Function* является самостоятельной процедурой, которая может получать аргументы, выполнять последовательность инструкций и изменять значения своих аргументов. Для возврата значения из функции следует присвоить значение имени функции. Любое число таких инструкций присвоения может находиться в любом месте функции среди управляющих инструкций.

Оператор вызова функции имеет *синтаксис*: ИмяФункции ([([СписокФактическихАргументов])[]).

Если функция не имеет аргументов, то круглые скобки необязательны. Но чтобы отличить оператор вызова функции от ссылки на переменную, рекомендуется в данном операторе всегда указывать круглые скобки.

1.17.7. Управляющие операторы

К управляющим операторам относятся условный оператор, оператор выбора, операторы цикла.

1.17.8. Условный оператор

Условный оператор задает выполнение определенных групп инструкций в зависимости от значения выражения.

Синтаксис: If Условие Then Инструкции [Else Инструкции_else].

Если *Условие* принимает значение *True* (истина), то выполняется инструкция (или инструкции) после *Then* (иначе), т. е. если *False* (ложь), то выполняется инструкция (или инструкции) после *Else*. Ветвь *Else* является необязательной. Если инструкций две или более, то они разделяются двоеточием.

Наиболее часто используется форма *синтаксиса* условного оператора в виде блока:

```

If Условие Then
    [Инструкции]
...
[ElseIf Условие-n Then
    [Инструкции_elseif]
[Else
    [Инструкции_else]]
End If

```

1.17.9. Оператор выбора

Оператор выбора используется в случае, если необходимо проверить несколько условий и оператор *If* может стать слишком сложным. Оператор выполняет одну из нескольких групп инструкций в зависимости от значения выражения.

Синтаксис:

```

Select Case выражение
Case СписокВыражений_1
    Инструкции_1
...
[Case СписокВыражений_n
    [Инструкции_n]]
[Case Else

```

[Инструкции_else]]
End Select

Инструкции_n (необязательная часть) – одна или несколько инструкций, выполняемых в том случае, если выражение совпадает с любым компонентом списка *СписокВыражений_n*.

Инструкции_else (необязательная часть) – одна или несколько инструкций, выполняемых в том случае, если выражение не совпадает ни с одним из предложений *Case*.

1.17.10. Операторы цикла

Синтаксис оператора *For Next*:

For Счетчик = Начало To Конец [Step шаг]
 Инструкции
 [Exit For]
 [Инструкции]
Next [Счетчик]

Оператор *For Next* повторяет выполнение группы инструкций указанное число раз, пока *Счетчик* изменяется от начального значения до конечного с указанным шагом. Если *Шаг* не указан, то он полагается равным 1. Альтернативный способ выхода из цикла предоставляет инструкция *Exit For*.

Синтаксис оператора *For Each Next*:

For Each Элемент In Группа
 Инструкции
 [Exit For]
 [Инструкции]
Next [Элемент]

Повторяет выполнение группы инструкций для каждого элемента массива или семейства. Его можно использовать, например, для суммирования значений диапазона ячеек. Альтернативный способ выхода из цикла предоставляет инструкция *Exit For*.

Синтаксис оператора *Do While*:

Оператор *Do While* имеет две формы.

1. Условие перед телом цикла:

Do While Условие
 Инструкции
 [Exit Do]
 [Инструкции]
Loop

Повторяет выполнение набора инструкций, пока *Условие* имеет значение *True*. Если *Условие* сразу имеет значение *False*, то тело цикла не выполняется ни разу. Альтернативный выход из цикла обеспечивает инструкция *Exit Do*.

2. Условие после тела цикла:

Do
 Инструкции
 [Exit Do]
 [Инструкции]
While Условие

Условие проверяется после выполнения инструкций, поэтому тело цикла выполнится, по крайней мере, один раз.

1.17.11. Оператор With

Оператор *With* выполняет последовательность инструкций над одиночным объектом или типом данных, определяемым пользователем. Он избавляет от использования большого количества повторений имени одного и того же объекта при работе с его свойствами и методами. Кроме того, он структурирует код.

Синтаксис:

With ИмяОбъекта
 Инструкции
End With

Инструкция *With* позволяет выполнить последовательность инструкций над указанным объектом, не повторяя задания имени объекта. Например, если имеются несколько свойств, которые необходимо изменить для одиночного объекта, то удобнее поместить инструкции присвоения свойств внутрь управляющей структуры *With*, указав ссылку на объект один раз, вместо того, чтобы ссылаться на объект при каждом присвоении его свойств.

Раздел 2. ИСПОЛЬЗОВАНИЕ ФУНКЦИЙ EXCEL В ЭКОНОМИЧЕСКИХ РАСЧЕТАХ

2.1. Совместное использование логических функций

Постановка задачи

Коммерсант получает комиссионные в зависимости от объема продаж по обычной или премиальной ставке. Комиссионные рассчитываются по следующему критерию:

ЕСЛИ Объем продаж \geq Нормированный объем продаж
И Объем расходов $<$ Нормированный объем расходов
ТО Объем продаж * Премиальная ставка;
ИНАЧЕ Объем продаж * Обычная ставка.

Решение 1

В ячейке D4 (рис. 2.1) находится формула

=ЕСЛИ(И(B4 \geq \$A\$2; C4 $<$ \$B\$2); B4*\$D\$2; B4*\$C\$2).

Решение 2

В ячейке D4 (рис. 2.1) находится формула

=ЕСЛИ(ИЛИ(B4 $<$ \$A\$2; C4 \geq \$B\$2); B4*\$C\$2; B4*\$D\$2).

Комиссионные рассчитываются по тому же критерию, записанному с помощью функции ИЛИ:

ЕСЛИ Объем продаж $<$ Нормированный объем продаж
ИЛИ Объем расходов \geq Нормированный объем расходов
ТО Объем продаж * Обычная ставка;
ИНАЧЕ Объем продаж * Премиальная ставка.

	А	В	С	Д
1	Нормированный объем продаж	Нормированный объем расходов	Обычная ставка	Премиальная ставка
2	1000	300	15%	25%
3	Дата	Объем продаж	Объем расходов	Комиссионные
4	1.02.06	1 200	350	180
5	2.02.06	1 000	295	250

Рис. 2.1. Расчетная таблица задачи

2.2. Учет доходов за обучение

Постановка задачи

Одной из статей дохода института являются денежные суммы, которые студенты платят за обучение. Институт может отнести на свои доходы только сумму за обучение в текущем месяце.

Студенты могут вносить оплату ежемесячно или на несколько месяцев вперед. В последнем случае оставшаяся сумма используется для оплаты следующих месяцев. Имеются также студенты, которые оплачивают учебу с задержкой на определенный срок. Ликвидация задолженности также относится на доходы текущего месяца.

Таким образом, в начале каждого месяца студент имеет либо остаток денег, либо долг за прошлый период. Институт фиксирует поступления денег за текущий месяц на свой счет и устанавливает для каждого студента сумму средств на обучение в документе, форма которого представлена на рис. 2.2.

Фамилия И. О.	На начало месяца		В текущем месяце			На конец месяца	
	остаток	долг	поступило	оплатить	доход института	остаток	долг
Бондарь А. Л.	500		2 000	800	=	=	=
Котова Р. О.		600		900	=	=	=
Лазутин А. Д.			800	800	=	=	=
Милевич Е. В.	1 600			1 200	=	=	=
Новицкий В. Н.		800	2 000	800	=	=	=
Юрченко Л. П.		500	1 500	1 200	=	=	=
Яновская О. А.	100		800	1 200	=	=	=
Итого	Σ	Σ	Σ	Σ	Σ	Σ	Σ

Рис. 2.2. Расчетная таблица задачи

Необходимо рассчитать следующие данные по приведенным ниже выражениям:

Доход института в текущем месяце =

ЕСЛИ Остаток на начало месяца + Поступило в текущем месяце >= Долг на начало месяца + Оплатить в текущем месяце
ТО Долг на начало месяца + Оплатить в текущем месяце;
ИНАЧЕ Остаток на начало месяца + Поступило в текущем месяце

Остаток на конец месяца =

ЕСЛИ Остаток на начало месяца + Поступило в текущем месяце >= Долг на начало месяца + Оплатить в текущем месяце
ТО Остаток на начало месяца + Поступило в текущем месяце – Доход института в текущем месяце;
ИНАЧЕ 0

Долг на конец месяца =

ЕСЛИ Остаток на начало месяца + Поступило в текущем месяце < Долг на начало месяца + Оплатить в текущем месяце
ТО Долг на начало месяца + Оплатить в текущем месяце – Доход института в текущем месяце;
ИНАЧЕ 0.

После расчета данных величин необходимо обеспечить переход на следующий месяц. Он заключается в переносе остатка и долга на конец текущего месяца в графы *Остаток* и *Долг* на начало следующего месяца на новом листе рабочей таблицы.

Решение

1. На листе *Январь* создать расчетную таблицу и ввести формулы согласно табл. 2.1. Результат приведен на рис. 2.3.

Таблица 2.1. Формулы для реализации задачи

Название столбца	Адрес ячейки	Формула
Доход в текущем месяце	F3	=ЕСЛИ(B3+D3>=C3+E3;C3+E3;B3+D3)
Остаток на конец месяца	G3	=ЕСЛИ(B3+D3>=C3+E3;B3+D3-F3;0)
Долг на конец месяца	H3	=ЕСЛИ(B3+D3<C3+E3;C3+E3-F3;0)

2. Создать копию листа *Январь*, присвоив ей имя *Февраль*.

3. На листе *Февраль* выделить диапазон ячеек, где находятся остаток и долг на конец месяца по студентам (итоги не выделять, т. е. G3:H9), и скопировать в буфер обмена.

4. Активизировать ячейку B3, содержащую остаток на начало месяца для первого студента.

	А	В	С	Д	Е	Ф	Г	Н
	На начало месяца	В текущем месяце	На конец месяца					
1	Фамилия И.О.	Остаток	Долг	Поступило	Оплатить	Доход	Остаток	Долг
3	Бондарь А.Л.	500		2 000	800	800	1 700	0
4	Котова Р.О.		600		900	0	0	1500
5	Лазутин А.Д.			800	800	800	0	0
6	Милевич Е.В.	1 600			1 200	1 200	400	0
7	Новицкий В.Н.		800	2 000	800	1 600	400	0
8	Юрченко Л.П.		500	1 500	1 200	1 500	0	200
9	Яновская О.А.	100		800	1 200	900	0	300
10	Итого	2 200	1 900	7 100	6 900	6 800	2 500	2 000

Рис. 2.3. Результат расчета за январь

5. Выполнить команду *Правка* → *Специальная вставка* и в появившемся диалоговом окне *Специальная вставка* установить опцию *Значения* (рис. 2.4).

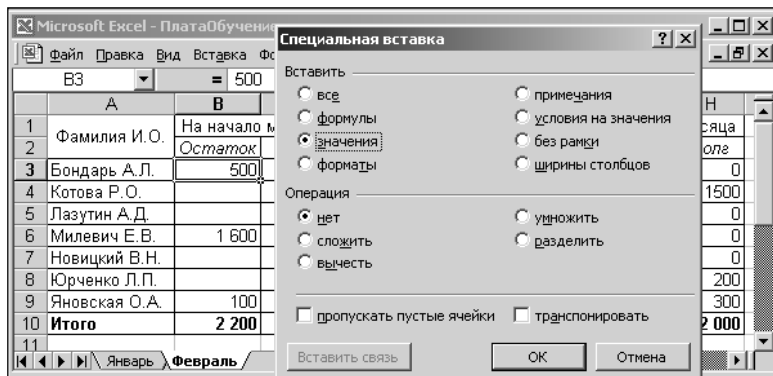


Рис. 2.4. Диалоговое окно *Специальная вставка*

6. Очистить диапазон ячеек, где содержатся данные по поступлениям и оплате для текущего месяца. Значения остатка и долга на конец и начало месяца должны совпасть (рис. 2.5).

7. Согласно табл. 2.2 ввести новые исходные данные в графы *Поступило в текущем месяце* и *Оплатить в текущем месяце* на листе *Февраль* (рис. 2.6) и выполнить переход на *Март*.

Microsoft Excel - ПлатОбучение											
Файл Правка Вид Вставка Формат Сервис Данные Окно Справка											
A1		= Фамилия И.О.									
A		B	C	D	E	F	G	H			
1	Фамилия И.О.	На начало месяца	В текущем месяце			На конец месяца					
2		Остаток	Долг	Поступило	Оплатить	Доход	Остаток	Долг			
3	Бондарь А.Л.	1 700	0			0	1 700	0			
4	Котова Р.О.	0	1 500			0	0	1500			
5	Лазутин А.Д.	0	0			0	0	0			
6	Милевич Е.В.	400	0			0	400	0			
7	Новицкий В.Н.	400	0			0	400	0			
8	Юрченко Л.П.	0	200			0	0	200			
9	Яновская О.А.	0	300			0	0	300			
10	Итого	2 500	2 000	0	0	0	2 500	2 000			
11											
Январь										Февраль	
Готово										NUM	

Рис. 2.5. Результат переноса данных на февраль

Таблица 2.2. Исходные данные за февраль

Фамилия, инициалы	Поступило	Оплатить
Бондарь А. Л.	0	800
Котова Р. О.	2 500	800
Лазутин А. Д.	700	700
Милевич Е. В.	0	1 000
Новицкий В. Н.	600	1 000
Юрченко Л. П.	0	1 100
Яновская О. А.	2 500	900

Microsoft Excel - ПлатежОбучение															
Файл Правка Вид Вставка Формат Сервис Данные Окно Справка															
A1		= Фамилия И.О.													
A		B		C		D		E		F		G		H	
1	Фамилия И.О.		На начало месяца			В текущем месяце				На конец месяца					
2			Остаток	Долг	Поступило	Оплатить	Доход	Остаток	Долг						
3	Бондарь А.Л.		1 700	0	0	800	800	900	0						
4	Котова Р.О.		0	1 500	2 500	800	2 300	200	0						
5	Лазутин А.Д.		0	0	700	700	700	0	0						
6	Милевич Е.В.		400	0	0	1 000	400	0	600						
7	Новицкий В.Н.		400	0	600	1 000	1 000	0	0						
8	Юрченко Л.П.		0	200	0	1 100	0	0	1300						
9	Яновская О.А.		0	300	2 500	900	1 200	1 300	0						
10	Итого		2 500	2 000	6 300	6 300	6 400	2 400	1 900						
11															
Январь										Февраль		Март			
Готово										NUM					

Рис. 2.6. Результат расчета за февраль

2.3. Использование функции СУММЕСЛИ

Постановка задачи

1. В табличном процессоре Excel необходимо создать расчетную таблицу согласно рис. 2.7.

Продажа валюты											
Дата	Курс у.е.	Пункт 1		Пункт 2		Пункт 3		Пункт 4		Итого	
		у.е.	руб.	у.е.	руб.	у.е.	руб.	у.е.	руб.	у.е.	руб.
01.02.06	2 170	520	=*	2350	=	5330	=	2330	=	ΣЕСЛИ*	ΣЕСЛИ
02.02.06	2 170	1390	=	790	=	1470	=	1260	=	ΣЕСЛИ	ΣЕСЛИ
03.02.06	2 175	1300	=	3750	=	3560	=	1400	=	ΣЕСЛИ	ΣЕСЛИ
04.02.06	2 170	1400	=	2570	=	3540	=	1210	=	ΣЕСЛИ	ΣЕСЛИ
05.02.06	2 180	740	=	680	=	3640	=	2480	=	ΣЕСЛИ	ΣЕСЛИ
06.02.06	2 175	860	=	570	=	890	=	3540	=	ΣЕСЛИ	ΣЕСЛИ
07.02.06	2 180	520	=	1020	=	2620	=	1170	=	ΣЕСЛИ	ΣЕСЛИ
	Итого:	Σ	Σ	Σ	Σ	Σ	Σ	Σ	Σ	Σ	Σ

Рис. 2.7. Расчетная таблица задачи

2. В ячейке с символами «=*» подсчитать графы *Продажа в рублях* по выражению

*Продажа в руб. = Продажа в у.е. * Курс у.е.*

3. Скопировать данную формулу в ячейки с символом «=».

4. В ячейке с символами «ΣЕСЛИ*» подсчитать итоговые значения по строкам с помощью функции СУММЕСЛИ.

5. Скопировать данную формулу в ячейки с символами «ΣЕСЛИ».

6. В ячейках с символом «Σ» подсчитать итоговые значения по столбцам.

7. Построить объемную трехмерную гистограмму продажи валюты в рублях за каждую дату по всем пунктам.

Решение

1. Расчет продажи в рублях приведен на рис. 2.8.

Продажа валюты											
Дата	Курс у.е.	Пункт 1		Пункт 2		Пункт 3		Пункт 4		Итого	
		у.е.	руб.	у.е.	руб.	у.е.	руб.	у.е.	руб.	у.е.	руб.
01.02.2006	2 170	520	1 128 400	2 350	5 099 500	5 330	11 566 100	2 330	5 056 100		
02.02.2006	2 170	1 390	3 016 300	790	1 714 300	1 470	3 189 900	1 260	2 734 200		
03.02.2006	2 175	1 300	2 827 500	3 750	8 156 250	3 560	7 743 000	1 400	3 045 000		
04.02.2006	2 170	1 400	3 038 000	2 570	5 576 900	3 540	7 681 800	1 210	2 625 700		
05.02.2006	2 180	740	1 613 200	680	1 482 400	3 640	7 935 200	2 480	5 406 400		
06.02.2006	2 175	860	1 870 500	570	1 239 750	890	1 935 750	3 540	7 699 500		
07.02.2006	2 180	520	1 133 600	1 020	2 223 600	2 620	5 711 600	1 170	2 550 600		
Итого											

Рис. 2.8. Результат расчета продажи в рублях

Продажа валюты											
Дата	Курс у.е.	Пункт 1		Пункт 2		Пункт 3		Пункт 4		Итого	
		у.е.	руб.	у.е.	руб.	у.е.	руб.	у.е.	руб.	у.е.	руб.
01.02.2006	2 170	520	1 128 400	2 350	5 099 500	5 330	11 566 100	2 330	5 056 100	10 530	22 850 100
02.02.2006	2 170	1 390	3 016 300	790	1 714 300	1 470	3 189 900	1 260	2 734 200	4 910	10 654 700
03.02.2006	2 175	1 300	2 827 500	3 750	8 156 250	3 560	7 743 000	1 400	3 045 000	10 010	21 771 750
04.02.2006	2 170	1 400	3 038 000	2 570	5 576 900	3 540	7 681 800	1 210	2 625 700	8 720	18 922 400
05.02.2006	2 180	740	1 613 200	680	1 482 400	3 640	7 935 200	2 480	5 406 400	7 540	16 437 200
06.02.2006	2 175	860	1 870 500	570	1 239 750	890	1 935 750	3 540	7 699 500	5 860	12 745 500
07.02.2006	2 180	520	1 133 600	1 020	2 223 600	2 620	5 711 600	1 170	2 550 600	5 330	11 619 400
Итого											

Рис. 2.9. Результат расчета итоговых значений

2.4. Использование смешанных ссылок

Постановка задачи

1. В табличном процессоре Excel нужно создать расчетную таблицу согласно рис. 2.10.
2. Графу *Доля акций в общем объеме* рассчитать по выражению:

$$\text{Доля акций в общем объеме (\%)} = \text{Количество акций} / \text{Итого по количеству акций}$$

Распределение прибыли акционерного общества

Акционер-ы	Количество акций	Доля акций в общем объеме	Январь	Февраль	Март	Апрель	Май	Июнь	Итого	
			Прибыль по месяцам							
			3 500 000	4 000 000	6 500 000	4 200 000	7 000 000	7 300 000		Σ
Факел	500	%	=*	=	=	=	=	=	Σ	
Марс	300	%	=	=	=	=	=	=	Σ	
Орбита	400	%	=	=	=	=	=	=	Σ	
Сож	300	%	=	=	=	=	=	=	Σ	
Турист	800	%	=	=	=	=	=	=	Σ	
Глобус	1 000	%	=	=	=	=	=	=	Σ	
Волна	200	%	=	=	=	=	=	=	Σ	
Итого	Σ	%	Σ	Σ	Σ	Σ	Σ	Σ	Σ	

Рис. 2.10. Расчетная таблица задачи

3. В ячейке с символами «=*» подсчитать значение прибыли акционера по выражению:

$$\text{Прибыль акционера} = \text{Доля акций в общем объеме (\%)} * \text{Прибыль за месяц}$$

4. Копировать данную формулу в ячейки с символом «=».
5. В ячейках с символом «Σ» подсчитать итоговые значения по строкам и столбцам.
6. Исправить формулу для расчета ежемесячной прибыли каждого акционера для округления результата до тысяч.
7. Построить накопительные графики для прибыли акционеров.
8. Построить накопительную диаграмму с областями для прибыли акционеров.

Решение

1. Расчет доли акций в общем объеме приведен на рис. 2.11.

Распределение прибыли акционерного общества									
Акционеры	Количество акций	Доля акций в общем объеме	Январь	Февраль	Март	Апрель	Май	Июнь	Итого
			3 500 000	4 000 000	6 500 000	4 200 000	7 000 000	7 300 000	
Факел	500	14,3%	500 000	571 429	928 571	600 000	1 042 857	4 642 857	
Марс	300	8,6%	300 000	342 857	557 143	360 000	600 000	2 785 714	
Орбита	400	11,4%	400 000	457 143	742 857	480 000	800 000	3 714 286	
Сож	300	8,6%	300 000	342 857	557 143	360 000	600 000	2 785 714	
Турист	800	22,9%	800 000	914 286	1 485 714	960 000	1 600 000	7 428 571	
Глобус	1 000	28,6%	1 000 000	1 142 857	1 857 143	1 200 000	2 000 000	9 285 714	
Волна	200	5,7%	200 000	228 571	371 429	240 000	400 000	1 857 143	
Итого	3 500	100,0%	3 500 000	4 000 000	6 500 000	4 200 000	7 000 000	32 500 000	

Рис. 2.11. Результат расчета доли акций в общем объеме

2. Расчет прибыли без округления до тысяч (рис. 2.12).
3. Расчет прибыли, округленной до тысяч (рис. 2.13).

Распределение прибыли акционерного общества									
Акционеры	Количество акций	Доля акций в общем объеме	Январь	Февраль	Март	Апрель	Май	Июнь	Итого
			3 500 000	4 000 000	6 500 000	4 200 000	7 000 000	7 300 000	
Факел	500	14,3%	500 000	571 429	928 571	600 000	1 042 857	4 642 857	
Марс	300	8,6%	300 000	342 857	557 143	360 000	600 000	2 785 714	
Орбита	400	11,4%	400 000	457 143	742 857	480 000	800 000	3 714 286	
Сож	300	8,6%	300 000	342 857	557 143	360 000	600 000	2 785 714	
Турист	800	22,9%	800 000	914 286	1 485 714	960 000	1 600 000	7 428 571	
Глобус	1 000	28,6%	1 000 000	1 142 857	1 857 143	1 200 000	2 000 000	9 285 714	
Волна	200	5,7%	200 000	228 571	371 429	240 000	400 000	1 857 143	
Итого	3 500	100,0%	3 500 000	4 000 000	6 500 000	4 200 000	7 000 000	32 500 000	

Рис. 2.12. Результат расчета без округления

Распределение прибыли акционерного общества									
А	В	С	Д	Е	Ф	Г	Н	И	Ж
Акцияры	Количество акций	Доля акций в общем объеме	Январь	Февраль	Март	Апрель	Май	Июнь	Итого
			3 500 000	4 000 000	6 500 000	4 200 000	7 000 000	7 300 000	32 500 000
Факел	500	14,3%	500 000	571 000	929 000	600 000	1 000 000	1 043 000	4 643 000
Марс	300	8,6%	300 000	343 000	557 000	360 000	600 000	626 000	2 786 000
Орбита	400	11,4%	400 000	457 000	743 000	480 000	800 000	834 000	3 714 000
Сож	300	8,6%	300 000	343 000	557 000	360 000	600 000	626 000	2 786 000
Турист	800	22,9%	800 000	914 000	1 486 000	960 000	1 600 000	1 669 000	7 429 000
Глобус	1 000	28,6%	1 000 000	1 143 000	1 857 000	1 200 000	2 000 000	2 086 000	9 286 000
Волна	200	5,7%	200 000	229 000	371 000	240 000	400 000	417 000	1 857 000
Итого	3 500	100,0%	3 500 000	4 000 000	6 500 000	4 200 000	7 000 000	7 301 000	32 501 000

Рис. 2.13. Результат расчета с округлением

2.5. Расчет итогов с условием

Постановка задачи

1. В табличном процессоре Excel создать расчетную таблицу согласно рис. 2.14.

Расчет комиссионных по продажам

Нормированный объем продаж	Нормированный объем расходов	Обычная ставка	Премияльная ставка
1000	300	15%	25%
Дата	Объем продаж	Объем расходов	Комиссионные
01.02.06	1 000	295	=
02.02.06	1 200	350	=
03.02.06	1 500	270	=
04.02.06	1 000	300	=
05.02.06	900	290	=
06.02.06	900	310	=
07.02.06	1 100	280	=
Всего	Σ	Σ	Σ
Итого	объем продаж >= нормированного	объем расходов < нормированного	по премияль- ной ставке
	Σ ЕСЛИ 1	Σ ЕСЛИ 2	Σ ЕСЛИ 3
Итого торговых дней	СЧЕТ	Итого премиальных дней	СЧЕТЕСЛИ

Рис. 2.14. Расчетная таблица задачи

2. В ячейках с символом «=» необходимо подсчитать комиссионные по следующему критерию:

ЕСЛИ Объем продаж >= Нормированный объем продаж;

И Объем расходов < Нормированный объем расходов;

ТО Объем продаж * Премияльная ставка;

ИНАЧЕ Объем продаж * Обычная ставка.

3. В ячейках с символом «Σ» подсчитать итоговые значения по столбцам.

4. В ячейках с символами «Σ ЕСЛИ» нужно подсчитать итоговые значения по столбцам согласно выражениям:

Σ ЕСЛИ 1 = Итого по дням, когда объем продаж >= Нормированный объем продаж;

Σ ЕСЛИ 2 = Итого по дням, когда объем расходов < Нормированный объем расходов;

Σ ЕСЛИ 3 = Итого по дням, когда комиссионные рассчитывались по премияльной ставке.

5. В ячейках с символами «СЧЕТ» и «СЧЕТЕСЛИ» нужно подсчитать итоговые значения согласно выражениям:

Итого торговых дней = Общее количество торговых дней;

Итого премиальных дней = Количество дней, когда комиссионные рассчитывались по премияльной ставке

Решение

1. Расчетная таблица с формулами приведена на рис. 2.15. Для того, чтобы рассчитать значения Σ ЕСЛИ 3 и Итого премиальных дней, добавлен вспомогательный столбец Премияльный день.
2. Результат расчета приведен на рис. 2.16.

	A	B	C	D	E
1	Расчет комиссионных по продажам				
2	Нормированный объем продаж	Нормированный объем расходов	Обычная ставка	Премияльная ставка	
3	1000	300	15%	25%	
4	Дата	Объем продаж	Объем расходов	Комиссионные	Премияльный день
5	01.02.06	1 000	295	=ЕСЛИ(И(B5>=\$A\$3; C5<\$B\$3);B5*\$D\$3; B5*\$C\$3)	=И(B5>=\$A\$3; C5<\$B\$3)
6	02.02.06	1 200	350		
7	03.02.06	1 500	270		
8	04.02.06	1 000	300		
9	05.02.06	900	290		
10	06.02.06	900	310		
11	07.02.06	1 100	280		
12	Всего	=СУММ(B5:B11)	=СУММ(C5:C11)	=СУММ(D5:D11)	
13	Итого	объем продаж >= нор- мированного	объем расходов < нормированного	по премияльной ставке	
14		=СУММЕСЛИ (B5:B11;">="&A3; B5:B11)	=СУММЕСЛИ (C5:C11;"<"&B3; C5:C11)	=СУММЕСЛИ (E5:E11;"ИСТИНА"; D5:D11)	
15	Итого торговых дней	=СЧЕТ(A5:A11)	Итого премияльных дней	=СЧЕТ(E5:E11; "ИСТИНА")	

Рис. 2.15. Расчетная таблица задачи с формулами

	A	B	C	D	E
1	Расчет комиссионных по продажам				
2	Нормированный объем продаж	Нормированный объем расходов	Обычные комиссионные	Премияльные комиссионные	
3	1000	300	15%	25%	
4	Дата	Объем продаж	Объем расходов	Комиссионные	Премияльный день
5	01.02.2006	1 000	295	250	ИСТИНА
6	02.02.2006	1 200	350	180	ЛОЖЬ
7	03.02.2006	1 500	270	375	ИСТИНА
8	04.02.2006	1 000	300	150	ЛОЖЬ
9	05.02.2006	900	290	135	ЛОЖЬ
10	06.02.2006	900	310	135	ЛОЖЬ
11	07.02.2006	1 100	280	275	ИСТИНА
12	Всего	7 600	2 095	1 500	ЛОЖЬ
13	Итого	объем продаж >= нор- мированного	объем расходов < нормированного	по премияльной ставке	
14		5800	1135	900	
15	Итого торговых	7	Итого премияльных	3	

Рис. 2.16. Результат расчета

2.6. Учет расходов и налогов с накоплением

Постановка задачи

1. Рассчитать налоги и прибыль предприятия по видам деятельности за текущий месяц и с накоплением с начала года. Расходы предприятия за январь приведены на рис. 2.17, а структура таблицы для расчета налогов – на рис. 2.18. Значения налогов округлить до целых с помощью функции ОКРУГЛ. Ставки налогов: НДС – 18 %, с/х – 1 %, прибыль – 30 %. Выражения для расчета налогов и прибыли по каждому виду следующие:

НДС = Доход * Ставка налога НДС

Налог с/х = (Доход – НДС) * Ставка налога с/х

Себестоимость = Итого по виду с листа расходов за месяц

Балансовая прибыль = Доход – (НДС + Налог с/х + Себестоимость)

ЕСЛИ Балансовая прибыль > 0,

ТО Налог на прибыль = Балансовая прибыль * Ставка налога на прибыль;

ИНАЧЕ Налог на прибыль = 0

ЕСЛИ Балансовая прибыль > 0,

ТО Чистая прибыль = Балансовая прибыль – Налог на прибыль;

ИНАЧЕ Чистая прибыль = 0.

Расходы за январь

Вид деятельности	Зарплата	Материалы	Комплектующие	Итого
Запись CD-дисков	1250	570	430	Σ
Прокат DVD	900	200	120	Σ
Сканирование	725	50	90	Σ
Ксерокопирование	1100	450	300	Σ
Итого	Σ	Σ	Σ	Σ

Рис. 2.17. Расходы предприятия за январь

Доходы и налоги за январь

Вид деятельности	Доход	НДС	Налог с/х	Себестоимость	Балансовая прибыль	Налог на прибыль	Чистая прибыль
Запись CD-дисков	3 570	=	=	=	=	=	=
Прокат DVD	1 350	=	=	=	=	=	=
Сканирование	1 125	=	=	=	=	=	=
Ксерокопирование	2 790	=	=	=	=	=	=
Итого	Σ	Σ	Σ	Σ	Σ	Σ	Σ

Рис. 2.18. Расчетная таблица для доходов и налогов за январь

2. Рассчитать налоги за февраль согласно данным, приведенным на рис. 2.19, а затем с помощью диалогового окна *Специальная вставка* создать накопительную ведомость расходов и налогов с начала года (т. е. за два месяца).

Расходы и доходы за февраль

Вид деятельности	Расходы			Доходы
	Зарплата	Материалы	Комплектующие	
Запись CD-дисков	1350	615	370	3 820
Прокат DVD	1050	175	150	1 720
Сканирование	685	40	80	910
Ксерокопирование	1220	470	350	2 850

Рис. 2.19. Исходные данные для расчета за февраль

3. Прodelать аналогичные операции с произвольными данными за март.

Решение

1. Создать структуру листов рабочей книги для расчета налогов за январь. Структуры листов *Налоги*, *Расходы* и *Прибыль* приведены на рисунках 2.20–2.22.

Рис. 2.20. Структура листа *Налоги*

Рис. 2.21. Структура листа *Расходы*

Рис. 2.22. Структура листа *Прибыль*

2. На листе с расходами рассчитать итоговые значения по столбцам и строкам (рис. 2.23).

	A	B	C	D	E
	Вид деятельности	Зарплата	Материалы	Комплектующие	Итого
1	Запись CD-дисков	1250	570	430	2250
2	Прокат DVD	900	200	120	1220
3	Сканирование	725	50	90	865
4	Ксерокопирование	1100	450	300	1850
5	Итого	3975	1270	940	6185

Рис. 2.23. Результат расчета расходов за январь

3. Перейти на лист со ставками налогов и в диалоговом окне *Присвоение имени*, вызываемом командой меню *Вставка* → *Имя* → *Присвоить* и приведенном на рис. 2.24, создать автоматические имена (Ставка_НДС, Ставка_с_х, Ставка_прибыль) для каждой ставки налога. Данная операция упростит ввод формул для вычисления налогов.

4. Перейти на лист с прибылью и заполнить вторую строку формулами согласно табл. 2.3. Вставка имени в формулу реализуется в диалоговом окне *Вставка имени*, вызываемом командой меню *Вставка* → *Имя* → *Вставить* и приведенном на рис. 2.25. Строки 3–5 рассчитываются копированием строки 2.

	A	B
1	Ставка НДС	18%
2	Ставка с/х	1%
3	Ставка прибыль	30%

Рис. 2.24. Диалоговое окно *Присвоение имени*

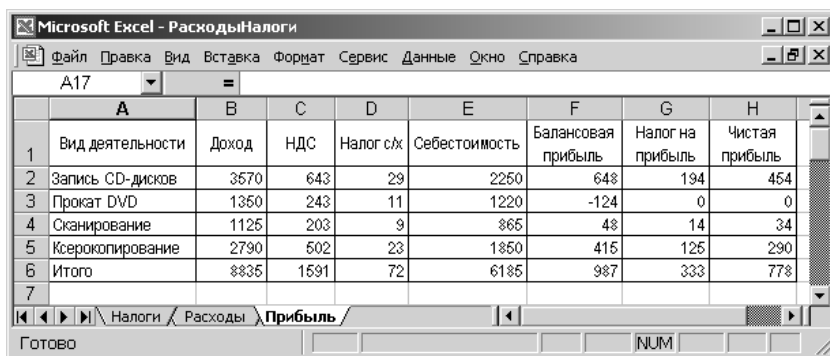
Таблица 2.3. Формулы для реализации задачи

Название столбца	Адрес ячейки	Формула
НДС	C2	=ОКРУГЛ(В2*Ставка_НДС;0)
Налог с/х	D2	=ОКРУГЛ((В2-C2)*Ставка_с_х;0)
Себестоимость	E2	=Расходы!E2
Балансовая прибыль	F2	=B2-(C2+D2+E2)
Налог на прибыль	G2	=ОКРУГЛ(ЕСЛИ(F2>0;F2*Ставка_прибыль;0);0)
Чистая прибыль	H2	=ЕСЛИ(F2>0;F2-G2;0)

	A	B	C	D	E	F	G	H
	Вид деятельности	Доход	НДС	Налог с/х	Себестоимость	Балансовая прибыль	Налог на прибыль	Чистая прибыль
1	Запись CD-дисков	3570						
2	Прокат DVD	1350						
3	Сканирование	1125						
4	Ксерокопирование	2790						
5	Итого							

Рис. 2.25. Диалоговое окно *Вставка имени*

5. Копировать формулы для других видов деятельности и подсчитать итоги по столбцам (рис. 2.26).



	A	B	C	D	E	F	G	H
	Вид деятельности	Доход	НДС	Налог с/х	Себестоимость	Балансовая прибыль	Налог на прибыль	Чистая прибыль
2	Запись CD-дисков	3570	643	29	2250	643	194	454
3	Прокат DVD	1350	243	11	1220	-124	0	0
4	Сканирование	1125	203	9	865	43	14	34
5	Ксерокопирование	2790	502	23	1850	415	125	290
6	Итого	8835	1591	72	6185	987	333	778

Рис. 2.26. Результат расчета налогов за январь

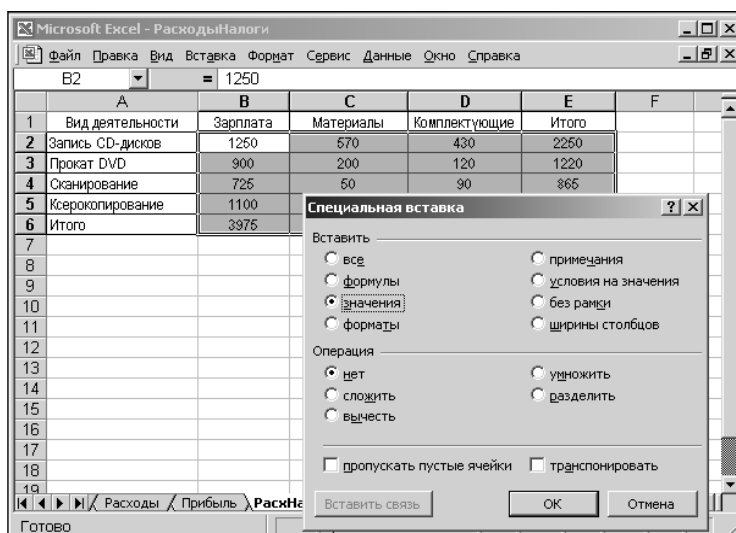
6. Так как накопленные расходы и прибыль с начала года совпадают с январскими расходами и прибылью, то создать копию листа *Расходы* с именем *РасхНакопл* и копию листа *Прибыль* с именем *ПрибНакопл*.

7. Чтобы дальше можно было накапливать данные (к остаткам прошлого периода добавлять текущие), нужно преобразовать все формулы на листах *РасхНакопл* и *ПрибНакопл* в значения, используя опцию *Значения* диалогового окна *Специальная вставка*, приведенного на рис. 2.27 и вызываемого командой меню *Правка* → *Специальная вставка* после копирования нужного диапазона в буфер обмена (B2:E6 на листе *РасхНакопл*, B2:H6 на листе *ПрибНакопл*).

8. Обновить листы *Расходы* и *Прибыль* данными за февраль (рисунки 2.28–2.29).

9. Обновить накопленные с начала года расходы и прибыль, добавив соответствующие данные за февраль, используя опции *Значения* и *Сложить* диалогового окна *Специальная вставка* после копирования нужного диапазона в буфер обмена (B2:E6 на листе *Расходы*, B2:H6 на листе *Прибыль*), как показано на рисунках 2.30–2.32.

10. Прodelать аналогичные операции с произвольными данными за март.



	A	B	C	D	E	F
	Вид деятельности	Зарплата	Материалы	Комплектующие	Итого	
2	Запись CD-дисков	1250	570	430	2250	
3	Прокат DVD	900	200	120	1220	
4	Сканирование	725	50	90	865	
5	Ксерокопирование	1100				
6	Итого	3975				

Рис. 2.27. Диалоговое окно *Специальная вставка*



	A	B	C	D	E
	Вид деятельности	Зарплата	Материалы	Комплектующие	Итого
2	Запись CD-дисков	1350	615	370	2335
3	Прокат DVD	1050	175	150	1375
4	Сканирование	685	40	80	805
5	Ксерокопирование	1220	470	350	2040
6	Итого	4305	1300	950	6555

Рис. 2.28. Результат расчета расходов за февраль

Вид деятельности	Доход	НДС	Налог с/х	Себестоимость	Балансовая прибыль	Налог на прибыль	Чистая прибыль
Запись CD-дисков	3820	688	31	2335	766	230	536
Прокат DVD	1720	310	14	1375	21	6	15
Сканирование	910	164	7	805	-66	0	0
Ксерокопирование	2850	513	23	2040	274	82	192
Итого	9300	1675	75	6555	995	318	743

Рис. 2.29. Результат расчета прибыли за февраль

Вид деятельности	Зарплата	Материалы	Комплектующие	Итого
Запись CD-дисков	1250	570	430	2250
Прокат DVD	900			
Сканирование	725			
Ксерокопирование	1100			
Итого	3975			

Рис. 2.30. Диалоговое окно Специальная вставка

Вид деятельности	Зарплата	Материалы	Комплектующие	Итого
Запись CD-дисков	2600	1185	800	4585
Прокат DVD	1950	375	270	2595
Сканирование	1410	90	170	1670
Ксерокопирование	2320	920	650	3890
Итого	8280	2570	1890	12740

Рис. 2.31. Результат расчета накопленных расходов

Вид деятельности	Доход	НДС	Налог с/х	Себестоимость	Балансовая прибыль	Налог на прибыль	Чистая прибыль
Запись CD-дисков	7390	1331	60	4585	1414	424	990
Прокат DVD	3070	553	25	2595	-103	6	15
Сканирование	2035	367	16	1670	-18	14	34
Ксерокопирование	5640	1015	46	3890	689	207	482
Итого	18135	3266	147	12740	1982	651	1521

Рис. 2.32. Результат расчета накопленных доходов

2.7. Использование формул массивов

2.7.1. Вычисление среднего отклонения

Постановка задачи

Даны выборки по двум измерениям. Необходимо вычислить среднее отклонение по каждой выборке с помощью обычных формул и с помощью формул массивов.

Решение

Результат расчета по обычной формуле приведен на рис. 2.33, а по формуле массива – на рис. 2.34.

	A	B	C	D
1	Вычисление среднего отклонения			
2	Выборка	Измерение1	Измерение2	Отклонение
3	Выборка1	84	87	3
4	Выборка2	75	73	-2
5	Выборка3	84	85	1
6	Выборка4	88	92	4
7	Выборка5	93	93	0
8	Выборка6	84	91	7
9	Выборка7	90	93	3
10				
11	Среднее отклонение:			2,285714286

Рис. 2.33. Результат расчета по обычной формуле

	A	B	C	D
1	Вычисление среднего отклонения			
2	Выборка	Измерение1	Измерение2	Отклонение
3	Выборка1	84	87	
4	Выборка2	75	73	
5	Выборка3	84	85	
6	Выборка4	88	92	
7	Выборка5	93	93	
8	Выборка6	84	91	
9	Выборка7	90	93	
10				
11	Среднее отклонение:			2,285714286

Рис. 2.34. Результат расчета по формуле массива

2.7.2. Вычисление квадратного корня

Постановка задачи

В диапазон A3:A6 введены числа: 6, 8, 9, 12. Необходимо вычислить квадратные корни этих чисел и поместить их соответственно в диапазон B3:B6, используя обычные формулы и формулы массива.

Решение

Результат расчета приведен на рис. 2.35.

	A	B	C	D
1	Вычисление квадратного корня			
2	Заданное число	Квадратный корень	Обычная формула	Формула массива
3	6	2,44949	=КОРЕНЬ(A3)	(=КОРЕНЬ(A3:A6))
4	8	2,828427	=КОРЕНЬ(A4)	(=КОРЕНЬ(A3:A6))
5	9	3	=КОРЕНЬ(A5)	(=КОРЕНЬ(A3:A6))
6	12	3,464102	=КОРЕНЬ(A6)	(=КОРЕНЬ(A3:A6))

Рис. 2.35. Результат расчета квадратного корня

2.7.3. Вычисление среднего положительных чисел

Постановка задачи

В диапазон A2:C4 введены данные, в которых содержатся положительные и отрицательные числа. Необходимо вычислить среднее из положительных чисел в этой группе.

Решение

Задача может быть решена с помощью формулы

$$=\text{СУММЕСЛИ}(\text{A2:C4};">0";\text{A2:C4})/\text{СЧЕТЕСЛИ}(\text{A2:C4};">0"),$$

где функция СУММЕСЛИ находит все положительные числа в диапазоне, а функция СЧЕТЕСЛИ вычисляет их количество (рис. 2.36).

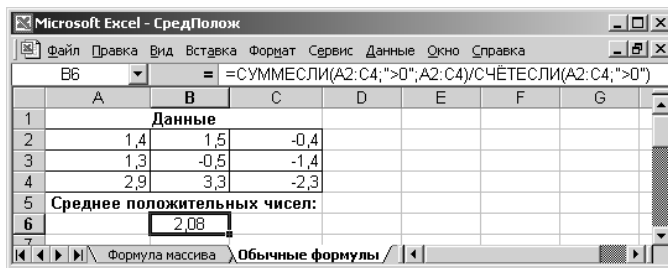


Рис. 2.36. Результат расчета по обычной формуле

Однако, более эффективным является использование следующей формулы массива (рис. 2.37):

$$\{=СРЗНАЧ(ЕСЛИ(A2:C4>0;A2:C4;""))\}.$$

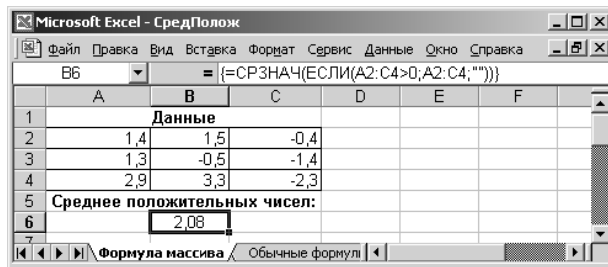


Рис. 2.37. Результат расчета с использованием формулы массива

В этом случае функция ЕСЛИ сравнивает каждый элемент введенной группы с нулем. Если элемент больше нуля, то функция ЕСЛИ возвращает это число, иначе она возвращает пустую строку. В результате получается массив, положительные элементы которого совпадают с соответствующими элементами исходного массива, а отрицательные элементы заменены пустой строкой. Функция СРЗНАЧ позволяет вычислить среднее значение элементов нового массива, а результат при этом возвращается в отдельную ячейку.

2.7.4. Транспонирование матрицы

Постановка задачи

В диапазон A2:B4 введены данные, образующие матрицу. Необходимо ее транспонировать (строки преобразовать в столбцы и наоборот).

Решение

Задача решается с помощью функции ТРАНСП (рис. 2.38). Ее результатом является прямоугольный диапазон. Поэтому начинать ввод функции необходимо с выделения этого диапазона (в данной задаче A7:C8), а завершить ввод функции (в ячейке, в строке формул или в диалоговом окне *Мастера функций*) следует так, как для формулы массивов, т. е. нажатием комбинации клавиш *Ctrl+Shift+Enter*.

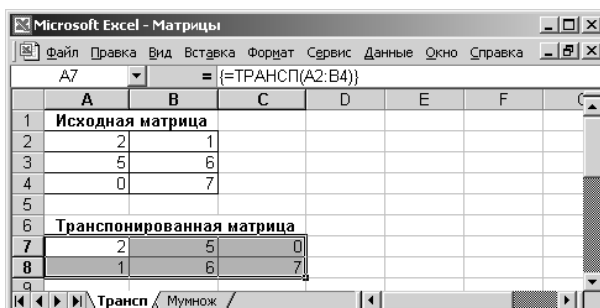


Рис. 2.38. Результат расчета транспонированной матрицы

2.7.5. Умножение матриц

Постановка задачи

В диапазоны A2:B4 и D2:G3 введены две матрицы. Необходимо найти их матричное произведение.

Решение

Задача решается с помощью функции МУМНОЖ (рис. 2.39). Ее аргументы – две матрицы, т. е. два прямоугольных диапазона. В матричном произведении количество столбцов левой матрицы должно совпадать с количеством строк правой матрицы. Результат умножения – также матрица, у которой количество строк равно количеству строк левой матрицы, а количество столбцов равно количеству столбцов правой матрицы.

	A	B	C	D	E	F	G
1	Исходная матрица 1			Исходная матрица 2			
2	2	1		3	5	1	0
3	5	6		-2	0	-5	1
4	0	7					
5							
6	Произведение матриц						
7	4	10	-3	1			
8	3	25	-25	6			
9	-14	0	-35	7			
10							

Рис. 2.39. Результат расчета умножения матриц

2.7.6. Умножение матрицы на транспонированный вектор

Постановка задачи

В диапазоны A2:C3 и A6:C6 введены, соответственно, матрица и вектор. Необходимо найти их матричное произведение.

Решение

Чтобы возможно было выполнить матричное произведение, вектор транспонируется. Результатом является вектор (рис. 2.40).

	A	B	C	D	E	F
1	Матрица					
2	2	1	-1			
3	0	3	1			
4					4	
5	Вектор				-2	
6	1	0	-2			
7						

Рис. 2.40. Результат расчета умножения матрицы на транспонированный вектор

2.8. Использование консолидации данных

Постановка задачи

Даны отчеты о продаже ноутбуков компьютерными фирмами «Сервер», «Рива» и «Компьютер Плаза» за три месяца – январь, февраль, март. Данные представлены в трех соответствующих файлах: *Сервер.xls*, *Рива.xls*, *КомпьютерПлаза.xls*. Однако списки включенных в них товаров несколько отличаются. Порядок перечисления товаров тоже различен (т. е. способы размещения информации в этих рабочих таблицах неидентичны). Требуется консолидировать информацию из всех рабочих книг в новую рабочую книгу *ИтогКонсолидация.xls*.

Решение

1. Создать новую папку *Консолидация*.
2. В табличном процессоре Excel создать рабочие книги *Сервер.xls*, *Рива.xls*, *КомпьютерПлаза.xls* и поместить их в папку *Консолидация*.
3. В рабочие книги ввести исходные данные согласно рисункам 2.41–2.43.

	A	B	C	D
1	Наименование Ноутбука	Январь	Февраль	Март
2	AcerAspire3023WLMi	21	20	15
3	AcerAspire5024WLMi	25	19	23
4	AcerAspire5002WLMi	19	15	11
5	AcerAspire3003WLM	15	16	20
6	HP Compaq nx6125	26	23	17
7	HP Compaq nc6120	11	11	6
8	HP Compaq nx6110	7	3	10
9	HP Compaq nx8220	4	0	5
10	HP Compaq tc4200	0	2	8
11	Samsung NP-P29	18	20	11

Рис. 2.41. Продажи компьютеров фирмой «Сервер»

	A	B	C	D
1	Наименование Ноутбука	Январь	Февраль	Март
2	AcerAspire5024WLMi	19	22	20
3	AcerAspire3023WLMi	20	10	17
4	AcerAspire3003WLM	16	18	17
5	AcerAspire5002WLMi	18	14	11
6	HP Compaq nx6125	18	12	16
7	HP Compaq nx6110	15	19	9
8	HP Compaq nc6120	11	17	11
9	HP Compaq tc4200	8	10	8
10	HP Compaq nx8220	10	0	15

Рис. 2.42. Продажи компьютеров фирмой «Рива»

	A	B	C	D
1	Наименование Ноутбука	Январь	Февраль	Март
2	AcerAspire3023WLMi	19	15	17
3	AcerAspire5002WLMi	20	12	25
4	AcerAspire5024WLMi	23	19	10
5	AcerAspire3003WLM	18	11	18
6	HP Compaq nx6125	22	21	19
7	HP Compaq nx6110	21	14	7
8	HP Compaq nc6120	8	7	11
9	HP Compaq tc4200	9	15	4
10	HP Compaq nx8220	5	1	9
11	Samsung NP-P29	10	22	15
12	Prestigio Nobile Sonoma	3	2	5

Рис. 2.43. Продажи компьютеров фирмой «КомпьютерПлаза»

4. Для консолидации информации необходимо создать новую рабочую книгу *ИтогКонсолидация.xls* и поместить ее в папку *Консолидация*.

5. В рабочей книге *ИтогКонсолидация.xls* установить курсор в ячейку A1 и выполнить команду *Данные* → *Консолидация*. На экране появится диалоговое окно *Консолидация*.

6. В поле *Функция* выбрать нужный тип консолидации. В данном примере – это функция *Сумма*.

7. В поле *Ссылка* ввести ссылку на диапазон рабочей таблицы файла *Сервер.xls*, воспользовавшись кнопкой *Обзор* и выбрав из появившегося диалогового окна необходимый файл. Ссылка должна включать адрес диапазона. Можно выбрать заведомо большой диапазон A1:D100, так как это дает гарантию правильной работы в случае добавления новых строк.

8. Нажать кнопку *Добавить* и ссылка попадет в список диапазонов [Сервер.xls]Лист1!\$A\$1:\$D\$100.

9. Аналогично ввести ссылки на диапазоны рабочих таблиц файлов *Рива.xls* и *КомпьютерПлаза.xls*: [Рива.xls]Лист1!\$A\$1:\$D\$100, [КомпьютерПлаза.xls]Лист1!\$A\$1:\$D\$100.

10. Поскольку способы размещения информации в рабочих таблицах различны, необходимо установить опции *Подписи верхней строки* и *Значения левого столбца*. В результате Excel будет подбирать данные по заголовкам.

11. Выбрать опцию *Создавать связи с исходными данными*. В результате Excel создаст структуру, содержащую внешние ссылки.

12. Для начала консолидации нажать *ОК*. Результат приведен на рис. 2.44.

1	2	A	B	C	D	E
				Январь	Февраль	Март
5	*	AcerAspire3023WLMi		60	45	49
9	*	AcerAspire5002WLMi		57	41	47
13	*	AcerAspire5024WLMi		67	60	53
17	*	AcerAspire3003WLM		49	45	55
21	*	HP Compaq nx6125		66	56	52
25	*	HP Compaq nx6110		43	36	26
29	*	HP Compaq nc6120		30	35	28
33	*	HP Compaq tc4200		17	27	20
37	*	HP Compaq nx8220		19	1	29
40	*	Samsung NP-P29		28	42	26
42	*	Prestigio Nobile Sonoma		3	2	5

Рис. 2.44. Результат консолидации

13. Отформатировать рабочую таблицу файла *ИтогКонсолидация.xls*, озаглавив столбцы и выполнив выравнивание (рис. 2.45).

14. Проанализировать результаты консолидации.

1	2	A	B	C	D	E
		Наименование Ноутбука	Наименование фирмы	Январь	Февраль	Март
2	*		Компьютерплаза	19	15	17
3	*		Рива	20	10	17
4	*		Сервер	21	20	15
5	*	AcerAspire3023WLMi		60	45	49
9	*	AcerAspire5002WLMi		57	41	47
13	*	AcerAspire5024WLMi		67	60	53
17	*	AcerAspire3003WLM		49	45	55
18	*		Компьютерплаза	22	21	19
19	*		Рива	18	12	16
20	*		Сервер	26	23	17
21	*	HP Compaq nx6125		66	56	52
25	*	HP Compaq nx6110		43	36	26
29	*	HP Compaq nc6120		30	35	28
33	*	HP Compaq tc4200		17	27	20
37	*	HP Compaq nx8220		19	1	29
40	*	Samsung NP-P29		28	42	26
42	*	Prestigio Nobile Sonoma		3	2	5

Рис. 2.45. Отформатированная рабочая таблица

Раздел 3. ПЛАНИРОВАНИЕ НА МАЛЫХ И СРЕДНИХ ПРЕДПРИЯТИЯХ СРЕДСТВАМИ EXCEL

3.1. Постановка задачи

Опыт показывает, что внедрение автоматизированных систем управления требует больших затрат, которые окупаются в лучшем случае за 3–4 года. Для малых и средних предприятий, которые называются профессиональными разработчиками систем электронной обработки данных, суммы просто неподъемны. В то же время потребность в автоматизации управления предприятием существует.

Между тем, имеется простое и удобное средство автоматизации управленческих расчетов – табличный процессор Excel. Малым и средним предприятиям не требуются большие и сложные базы данных. Данные могут быть расположены на листах в книгах Excel. Все необходимые вычисления можно выполнить, используя простые средства Excel, включая его функции и надстройки.

Работа по автоматизации управленческих расчетов может быть выполнена самостоятельно и, что немаловажно, постепенно. Это не требует больших затрат. Обучение работе с системой происходит автоматически, в процессе ее создания. Все, что для этого нужно, – это освоить технику использования Excel в управленческих расчетах и определить очередность автоматизации задач управления.

Традиционно на больших предприятиях в первую очередь автоматизируются задачи учета – бухгалтерского, коммерческого и производственного. Возможно, при разработке больших автоматизированных систем управления предприятием так и нужно поступать. Но учет – это только одна из четырех фаз управления: учет, анализ, прогноз, планирование. Причем структурирующей всю систему фазой является фаза планирования. Учитывать следует только те показатели, на основе которых принимаются решения, т. е. составляются планы. То же самое можно сказать и о прогнозировании финансово-экономических показате-

телей. Анализ – это прежде всего сравнение факта с планом, планирование и здесь определяет круг показателей, которые надо анализировать. Поэтому по крайней мере на малых и средних предприятиях в первую очередь надо автоматизировать именно плановые расчеты.

Продолжительность интервала среднесрочного планирования обычно устанавливают равной одному году. Это вызвано следующими причинами:

- каждое предприятие обязано сдавать в налоговую инспекцию годовой баланс;
- в акционерных обществах обязательно проведение годового собрания акционеров, на котором отчитывается руководство акционерного общества по итогам работы за год;
- практика ведения хозяйственных операций и обычаи предусматривают заключение многих договоров на год (аренда, договор поставки и т. п.).

Описание примера

В качестве примера используется несложное производство автоматов для продажи газированной воды. Чтобы пример не был слишком громоздким и не отвлекал от его основной задачи – построения системы плановых расчетов, – несложное само по себе предприятие еще более упрощено: в номенклатуре продукции оставлены всего два вида автоматов АК-3 и АТ-1000. Нормы времени, нормы расхода материалов, цены и другие обычно являются коммерческой тайной, поэтому численные значения норм, нормативов и других показателей несколько условны. Все это сделано для того, чтобы пример был максимально простым и наглядным.

На предприятии работают 112 человек (формально это предприятие можно отнести и к среднему). Из них 20 человек – инженерно-технический персонал и служащие (ИТР), включая директора предприятия. Из рабочих – 16 повременщиков и 76 сдельщиков, занятых в основном производстве.

Все эти люди располагаются в двух небольших зданиях (корпус 1 и корпус 2), которые являются собственностью предприятия и находятся на его балансе. В состав основных производственных фондов (ОПФ) предприятия также входят (оборудование (станки токарно-винторезные РТ2505, пресс П-511 Б, станки фрезерные 6610, станки фрезерные 6Р83Г) и прочие фонды (автокары, автомобили и т. д.).

Предприятие закупает материалы и комплектующие: лист стальной, уголки, цветные металлы, электрооборудование, краску, лак. Готовые изделия продаются в розницу самим предприятием.

Численные значения показателей, цен, норм и нормативов приводятся в процессе изложения материала.

Необходимо в табличном процессоре Excel автоматизировать решение следующих задач планирования:

- формирование проекта плана производства и продаж;
- расчет потребности в мощности и материалах;
- расчет численности сотрудников и фонда заработной платы;
- расчет себестоимости товарной продукции;
- расчет прибыли и рентабельности.

Организация данных

Все данные, необходимые для автоматизации плановых расчетов на малом или среднем предприятии, размещаются в книгах Excel. Организовать данные можно по-разному. Все данные для расчета плановых показателей малого предприятия можно разместить в одной книге Excel. С другой стороны, для каждой группы однородных данных можно создать свою книгу. Это оправданно, если объем исходной информации измеряется в тысячах записей, которые обрабатываются разными сотрудниками.

Так как в данном примере объем исходной информации небольшой, то используется единственная книга, названная *РасчетПлана.xls*. Состав и назначение листов приведено в табл. 3.1.

Таблица 3.1. Состав и назначение листов рабочей книги

Лист	Назначение
Прогноз	Прогноз начального варианта плана производства
КритОбъем	Расчет критического объема выпуска
ФондВрСтан	Плановый фонд времени работы оборудования
ФондВрРучн	Плановый фонд времени рабочих на ручных операциях
Числ	Численность – план по труду
НорВр	Нормы времени на станочные и ручные операции
Произв	Объем производства продукции и цен на нее
Мощн	Потребность в мощности
НорМат	Нормы расхода материалов и цены на них
Матер	Потребность в материалах

Лист	Назначение
Тариф	Часовые тарифные ставки и оклады
ФЗП	Фонд зарплаты
ОПФ	Стоимость и амортизация ОПФ
Себест	Расчет себестоимости
Прибыль	Расчет прибыли

3.2. Предплановые расчеты

3.2.1. Построение начального варианта плана производства

Формальной процедуры построения начального варианта плана производства не существует, т. е. принятие решения об объеме производства – это принятие рискованного решения. Риск несколько уменьшается, если предприятие работает не «про запас», а «на заказ». И в том и в другом случае полезен прогноз объема продаж. Часто этот прогноз делается по временному ряду продаж за несколько последних предплановых периодов.

Алгоритмы прогнозирования по временным рядам хорошо разработаны. Но всегда следует иметь в виду, что на реальную величину спроса влияет множество неконтролируемых или слабо контролируемых факторов, таких как сервис, послепродажное обслуживание, мода, предрассудки и предубеждения потребителей и т. п.

В Excel имеется функция ТЕНДЕНЦИЯ из категории *Статистические*, предназначенная для прогноза показателя (в данном случае объема продаж на некоторый товар) по временному ряду. Функция ТЕНДЕНЦИЯ строит прогноз на основе модели

$$y = a * x + b,$$

где a и b – параметры, вычисляемые функцией;

y – прогнозируемая переменная (в данном случае объем продаж);

x – независимая переменная (в данном случае номер планируемого года).

В рассматриваемой задаче прогноз объема плана продаж реализован на листе *Прогноз*. Структура данных приведена на рис. 3.1.

	А	В
1	Динамика продаж	
2	Год	Объем продаж
3	1	153 614
4	2	184 992
5	3	200 163
6	4	216 734
7	5	225 404
8		Прогноз
9	6	

Рис. 3.1. Структура данных на листе *Прогноз*

В контексте решаемой задачи для функции ТЕНДЕНЦИЯ необходимы следующие аргументы:

- известные значения y – объемы продаж за пять предплановых лет;
- известные значения x – пять предплановых лет, пронумерованных от 1 до 5;
- новое значение x – это номер планового года (в данном случае 6).

Результат вычислений показан на рис. 3.2.

	А	В	С
1	Динамика продаж		
2	Год	Объем продаж	
3	1	153 614	
4	2	184 992	
5	3	200 163	
6	4	216 734	
7	5	225 404	
8		Прогноз	
9	6	248 778	

Рис. 3.2. Результат прогнозирования

3.2.2. Расчет критического объема выпуска

Критический объем выпуска – это минимальный объем реализации, покрывающий минимальные необходимые затраты. Критический объем выпуска определяется по следующей формуле:

$$B_{кр} = ((1 + \alpha) * 3\Phi + Z_{пост} + Д) / (1 - n - \beta - \gamma * (1 + \alpha)),$$

где $B_{кр}$ – критический объем выпуска;

3Φ – фиксированная зарплата;

$Z_{пост}$ – условно-постоянные затраты за вычетом фиксированной зарплаты;

$Д$ – дополнительная нагрузка, включающая налог на имущество и другие налоги и обязательные фискальные платежи, облагаемой базой которых не является выручка (земельные платежи, погашение недоимки по налогам и другим обязательным платежам, штрафы, пени и т. д.), а также погашение кредита, проценты за кредит и другое, т. е. затраты, не зависящие от выручки;

α – нормативы отчислений во внебюджетные фонды и подоходного налога (единый социальный налог);

β – доля условно-переменных затрат за вычетом зарплаты в выручке;

γ – доля сдельной зарплаты в выручке;

n – суммарный норматив налогов, облагаемой базой которых является объем выпуска.

Параметры n и α устанавливаются государственными органами, и предприятие на них не влияет. Величины параметров β и γ зависят в основном от материало- и энергоемкости и от трудоемкости продукции, соответственно.

Расчет критического объема выпуска реализован на листе *КритОбъем*. Структура расчетной таблицы приведена на рис. 3.3, результат вычислений – на рис. 3.4.

	А	В	С
1	Расчет критического объема выпуска		
2	ПОКАЗАТЕЛЬ	Ед. измер.	Значение
3	Фиксированная зарплата	тыс. руб.	168 000
4	Условно-постоянные затраты за вычетом фиксированной зарплаты	тыс. руб.	272 000
5	Дополнительная нагрузка	тыс. руб.	1 120
6	Ставка единого социального налога	доля	0,365
7	Доля условно-переменных затрат за вычетом зарплаты в выручке	доля	0,82
8	Доля сдельной зарплаты в выручке	доля	0,01
9	Суммарный норматив налогов, облагаемой базой которых является объем выпуска	доля	0
10	Критический объем выпуска	тыс. руб.	

Рис. 3.3. Структура данных на листе *КритОбъем*

	А	В	С
1	Расчет критического объема выпуска		
2	ПОКАЗАТЕЛЬ	Ед. измер.	Значение
3	Фиксированная зарплата	тыс. руб.	168 000
4	Условно-постоянные затраты за вычетом фиксированной зарплаты	тыс. руб.	272 000
5	Дополнительная нагрузка	тыс. руб.	1 120
6	Ставка единого социального налога	доля	0,365
7	Доля условно-переменных затрат за вычетом зарплаты в выручке	доля	0,82
8	Доля сдельной зарплаты в выручке	доля	0,01
9	Суммарный норматив налогов, облагаемой базой которых является объем выпуска	доля	0
10	Критический объем выпуска	тыс. руб.	3 020 379

Рис. 3.4. Результат расчета критического объема выпуска

3.3. Планирование наличия мощности

3.3.1. Расчет действительного фонда времени работы оборудования

Расчеты наличия производственных и финансовых ресурсов необходимы для обоснования плана производства и реализации продукции. При проведении этих расчетов учитываются основные виды ресурсов: оборудование, персонал, материалы (включая инструмент и оснастку), ликвидные активы (денежные средства и быстрореализуемые ценные бумаги).

Действительный (плановый) фонд времени работы оборудования – показатель объема такого производственного ресурса, как оборудование. Расчет действительного фонда времени работы оборудования состоит из нескольких этапов:

1. Расчет действительного фонда времени работы станка в каждой группе.
2. Расчет суммарного действительного фонда времени работы оборудования по каждой группе.

3. Корректировка расчетов с учетом коэффициентов сменности, выполнения норм и использования оборудования.

Оценкой объема основных производственных фондов для целей среднесрочного планирования служит фонд времени работы оборудования, измеряемый обычно в станко-часах.

Различают следующие фонды времени работы оборудования: календарный (K), номинальный (H), действительный в днях ($Дд$) и часах ($Дч$), плановый ($П$). Для их вычисления используются следующие выражения:

K = Количество календарных дней в плановом периоде;

$H = K -$ Количество выходных и праздничных дней;

$Дд = H -$ Количество дней на ремонт оборудования;

$Дч = Дд * \text{Продолжительность смены} * \text{Количество смен};$

$П = Дч * \text{Количество единиц оборудования}.$

Расчет фонда времени работы оборудования реализован на листе *ФондВрСтан*. Структура расчетной таблицы приведена на рис. 3.5, результат вычислений – на рис. 3.6.

	A	B	C	D	E	F	G
1	Календарный фонд	Выходные и праздники	Номинальный фонд рабочего времени, дни	Номинальная продолжительность смены, ч	Льготное сокращение смены	Плановая продолжительность смены, ч	
2	365	109		8	0,2		
3							
4	Расчет планового фонда времени работы оборудования						
5	Наименование оборудования	Капитальный и профилактический ремонт	Действительный фонд времени работы единицы оборудования, дни	Количество смен	Действительный фонд времени работы единицы оборудования, ч	Количество единиц оборудования	Плановый фонд времени, всего
6	Станок токарно-винторезный PT2505	23		1,4		7	
7	Пресс П-511Б	19		1		16	
8	Станок фрезерный 6610	21		1		5	
9	Станок фрезерный 6P83Г	21		1		20	

Рис. 3.5. Структура данных листа *ФондВрСтан*

	A	B	C	D	E	F	G
1	Календарный фонд	Выходные и праздники	Номинальный фонд рабочего времени, дни	Номинальная продолжительность смены, ч	Льготное сокращение смены	Плановая продолжительность смены, ч	
2	365	109	256	8	0,2	7,8	
3							
4	Расчет планового фонда времени работы оборудования						
5	Наименование оборудования	Капитальный и профилактический ремонт	Действительный фонд времени работы единицы оборудования, дни	Количество смен	Действительный фонд времени работы единицы оборудования, ч	Количество единиц оборудования	Плановый фонд времени, всего
6	Станок токарно-винторезный PT2505	23	233	1,4	2 544	7	17 811
7	Пресс П-511Б	19	237	1	1 849	16	29 578
8	Станок фрезерный 6610	21	235	1	1 833	5	9 165
9	Станок фрезерный 6P83Г	21	235	1	1 833	20	36 660

Рис. 3.6. Результат расчета фонда времени работы оборудования

3.3.2. Расчет планового фонда времени рабочих

Плановый фонд времени одного рабочего-станочника равен плановому фонду времени соответствующего станка при односменном режиме работы ($K_{см} = 1$) и с учетом многостаночного обслуживания. В нашем примере многостаночного обслуживания нет, поэтому ни отдельного расчета, ни отдельной таблицы не требуется.

Расчет планового фонда времени рабочего (иногда этот расчет называют балансом рабочего времени) может проводиться для среднесписочного рабочего или для среднего рабочего данной профессии. Для построения сбалансированного по ресурсам плана целесообразно строить не агрегированный баланс рабочего времени, а баланс рабочего времени в разрезе профессий. Агрегированный баланс, конечно, менее трудоемок, но использование вычислительной техники снимает эту проблему.

В столбце А на рис. 3.7 перечислены показатели, используемые в этом расчете. Календарный (K) и номинальный (H) фонды времени те же, что и в расчете фонда времени оборудования, – количество дней в году и количество рабочих дней в году, соответственно. Действительный фонд времени равен номинальному за вычетом планируемых неявок на работу (отпуска, выполнение государственных или обществен-

ных обязанностей и т. п.). Действительный фонд времени вычисляется сначала в днях ($Dд$), затем в часах ($Dч$), при этом учитывается льготное сокращение рабочего дня в предпраздничные дни. Итоговый плановый ($П$) фонд рабочего времени рабочих данной профессии (или всех рабочих) определяется как произведение действительного фонда времени одного рабочего на плановую численность. Таким образом, используются следующие выражения:

K = Количество календарных дней в плановом периоде;

$H = K -$ Количество выходных и праздничных дней;

$Dд = H -$ Сумма планируемого отсутствия на работе;

$ПП = НП - ЛС$;

$Dч = Dд * ПП$;

$П = Dч * Численность$,

где $НП$ – номинальная продолжительность рабочего дня;

$ЛС$ – льготное сокращение рабочего дня;

$ПП$ – плановая продолжительность рабочего дня.

Расчет планового фонда времени рабочих на ручных операциях реализован на листе *ФондВрРучн*. Структура расчетной таблицы приведена на рис. 3.7. Численность рабочих находится на листе *Числ* (рис. 3.8), занимая два несмежных диапазона. Значит, для отображения численности на листе *ФондВрРучн* следует дважды воспользоваться ссылкой на соответствующие ячейки листа *Числ*.

	А	В	С	Д	Е	Ф
1	Календарный фонд	Выходные и праздники	Номинальный фонд рабочего времени, дни	Номинальная продолжительность смены, ч	Льготное сокращение смены	Плановая продолжительность смены, ч
2	365	109		8	0,2	
3	Плановые неявки на работу	Очередные отпуска	Дополнительные отпуска	Отпуска по временной нетрудоспособности	Действительный фонд рабочего времени, дни	Действительный фонд рабочего времени, ч
4	4	30	0	4		
5	Наименование профессии	Численность	Итоговый плановый фонд			
6	Слесарь-сборщик					
7	Наладчик					
8	Контролер					
9	Слесарь-ремонтник					
10	Электрик					
11	Кладовщик					
12	Уборщица					
13	Грузчик					
14	Водитель					
15	Водитель автокара					

Рис. 3.7. Структура данных листа *ФондВрРучн*

	А	В	С	Д	Е	Ф	Г
1	План по труду						
2	Основные производственные рабочие (сдельщики)			Вспомогательные рабочие (повременщики)		ИТР и служащие	
3	Наименование профессии	Численность расчетная	Численность плановая	Наименование профессии	Численность плановая	Наименование должности	Численность плановая
4	Токарь			Слесарь-ремонтник	2	Директор	1
5	Штамповщик			Электрик	3	Зам. директора по сбыту	1
6	Фрезеровщик			Кладовщица	2	Гл. инженер	1
7	Фрезеровщик			Уборщица	4	Гл. бухгалтер	1
	Слесарь-сборщик			Грузчик	1	Менеджер	7
8	Наладчик			Водитель	3	Мастер	7
9	Контролер			Водитель автокара	1	Бухгалтер	2
10	Итого сдельщики			Итого повременщики		Итого ИТР и служащие	
11	Численность списочная всего						

Рис. 3.8. Структура данных листа *Числ*

Текущий результат расчета планового фонда времени рабочих на ручных операциях приведен на рис. 3.9. В нем численность и плановый фонд части производственных рабочих равны нулю. Очевидно, они будут автоматически изменены после ввода формул на лист *Числ*.

	А	В	С	Д	Е	Ф
1	Календарный фонд	Выходные и праздники	Номинальный фонд рабочего времени, дни	Номинальная продолжительность смены, ч	Льготное сокращение смены	Плановая продолжительность смены, ч
2	365	109	256	8	0,2	7,8
3	Плановые неявки на работу	Очередные отпуска	Дополнительные отпуска	Отпуска по временной нетрудоспособности	Действительный фонд рабочего времени, дни	Действительный фонд рабочего времени, ч
4	4	30	0	4	218	1 700,4
5	Расчет планового фонда времени рабочих на ручных операциях					
6	Наименование профессии	Численность	Итоговый плановый фонд			
7	Слесарь-сборщик	0	0			
8	Наладчик	0	0			
9	Контролер	0	0			
10	Слесарь-ремонтник	0	0			
11	Электрик	3	5 101,2			
12	Кладовщик	2	3 400,8			
13	Уборщица	4	6 801,6			
14	Грузчик	1	1 700,4			
15	Водитель	3	5 101,2			
16	Водитель автокара	1	1 700,4			

Рис. 3.9. Результат расчета планового фонда времени рабочих на ручных операциях

3.4. Планирование потребности в мощности

План производства прежде всего должен быть реалистичным, т. е. потребность в производственной мощности не должна превышать ее наличия. *Наличная мощность* – это плановые (действительные) фонды времени работы оборудования и основных производственных рабочих на ручных операциях. *Производственная мощность* – не число, а вектор. Размерность этого вектора равна сумме количества групп взаимозаменяемого оборудования и количества профессий основных производственных рабочих на ручных операциях. Различные векторы мощности по-прежнему несравнимы, но для оценки сбалансированности потребности в мощности с ее плановым наличием этого и не требуется. Для построения реалистичного плана требуется сравнивать потребность в мощности с ее плановым наличием.

Итак, далее термин *мощность* используется в следующем значении. *Плановая наличная мощность (плановое наличие мощности)* – это вектор, элементами которого являются плановые фонды времени групп взаимозаменяемого оборудования и профессий основных производственных рабочих на ручных операциях. Плановая потребность в мощности на производственную программу – это вектор той же размерности (для всех операций за исключением ремонтных), рассчитываемый по формуле

$$ПМ_i = \sum_j HB_{ij} * X_j / 100,$$

где i – номер (шифр, код) группы взаимозаменяемого оборудования или профессии основных производственных рабочих на ручных операциях;

j – номер (шифр, код) изделия;

$ПМ_i$ – потребность в мощности номер i ;

HB_{ij} – норма времени i -ой группы взаимозаменяемого оборудования или профессии основных производственных рабочих на ручных операциях на единицу изделия j (установлены на 100 изделий, поэтому результат умножения нужно еще разделить на 100);

x_j – плановый объем производства изделия j .

То же самое уравнение плановой потребности в мощности имеет тривиальный вид в матричной форме:

$$ПМ = нв * x / 100,$$

где $ПМ$ – вектор потребности в мощности;

$нв$ – матрица норм времени;

x – вектор объема производства продукции.

В Excel есть функция МУМНОЖ, предназначенная для умножения матриц. Аргументы данной функции – это две матрицы (любой из них может быть вектор, как частный случай матрицы). Результатом функции МУМНОЖ является прямоугольный диапазон. Поэтому начинать ввод функции необходимо с

выделения этого диапазона, а завершить ввод функции (в ячейке, в строке формул или в диалоговом окне *Мастера функций*) следует так, как для формулы массивов, т. е. нажатием комбинации клавиш *Ctrl+Shift+Enter*.

Нормы времени находятся на листе *НорВр*, структура которого приведена на рис. 3.10. План производства продукции (или проект плана) задается на листе *Произв*, пример которого показан на рис. 3.11. Расчетная таблица для потребности в мощности находится на листе *Мощность* (рис. 3.12).

	A	B	C	D	E
1	Нормы времени на станочные и ручные операции				
2	Наименование оборудования/профессии	Ед. изм.	Изделие		Ремонтосложность (ЕРС)
3			АК-3	АТ-1000	
4	Станок токарно-винторезный РТ2505	ч/100 шт.	6	6	3
5	ПресскП-511Б	ч/100 шт.	7,5	7,8	5
6	Станок фрезерный 6610	ч/100 шт.	2,7	3,6	14
7	Станок фрезерный 6Р83Г	ч/100 шт.	11,1	11	14
8	Слесарь-сборщик	ч/100 шт.	6	6	
9	Наладчик	ч/100 шт.	2,5	2,8	
10	Контролер	ч/100 шт.	0,9	1,1	
11	Слесарь-ремонтник	ч/ЕРС			6

Рис. 3.10. Структура данных листа *НорВр*

	A	B	C	D
1	План производства готовой продукции			
2	Наименование изделия	Ед. изм.	Количество	Цена
3	АК-3	шт.	240 000	140 000
4	АТ-1000	шт.	110 000	130 400

Рис. 3.11. Структура данных листа *Произв*

	A	B	C	D	E
1	Баланс мощности				
2	Наименование группы оборудования или профессии	Ед. изм.	Потребность в мощности	Наличие мощности	Отношение потребности к наличию
3	Станок токарно-винторезный РТ2505	станко-ч			
4	ПресскП-511Б	станко-ч			
5	Станок фрезерный 6610	станко-ч			
6	Станок фрезерный 6Р83Г	станко-ч			
7	Слесарь-сборщик	нормо-ч			
8	Наладчик	нормо-ч			
9	Контролер	нормо-ч			
10	Слесарь-ремонтник	нормо-ч			

Рис. 3.12. Структура данных листа *Мощность*

Анализируя данные исходной таблицы *Нормы времени на станочные и ручные операции*, приведенной на рис. 3.10, можно сделать вывод, что в столбце *Потребность в мощности* таблицы *Баланс мощности* (рис. 3.12) в результате матричного умножения будет заполнен диапазон С3:С9.

Потребность в мощности ремонтников (строка 10 таблицы *Баланс мощности*) вычисляется другим способом. Нормы времени на ремонтные работы устанавливаются на единицу ремонтосложности (ЕРС). В рамках единой системы планово-предупредительного ремонта каждому станку (агрегату) присвоено определенное количество условных единиц ремонтосложности, характеризующих конструктивно-технологическую сложность ремонта данного станка по сравнению со станком-эталонем. Данные о ремонтосложности оборудования, введенные на лист *НорВр*, приведены на рис. 3.10. Потребность в мощности ремонтников для данного станочного парка рассчитывается по формуле

$$ПМр = нвр * кол * рс,$$

где *ПМр* – потребность в мощности ремонтников;

нвр – норма времени на единицу ремонтосложности (ячейка Е11 на листе *НорВр*);

кол – вектор количества единиц оборудования (лист *ФондВрСтан*);

рс – вектор ремонтосложности оборудования.

Так как оба вектора *кол* и *рс* находятся в столбцах, то для нахождения их скалярного произведения можно воспользоваться функцией СУММПРОИЗВ, аргументами которой являются эти векторы. Результат вычисления потребности в мощности приведен на рис. 3.13.

	A	B	C	D	E
1	Баланс мощности				
2	Наименование группы оборудования или профессии	Ед. изм.	Потребность в мощности	Наличие мощности	Отношение потребности к наличию
3	Станок токарно-винторезный РТ2505	станко-ч	21 000	17 811	1,2
4	ПрессП-511Б	станко-ч	26 580	29 578	0,9
5	Станок фрезерный 6610	станко-ч	10 440	9 163	1,1
6	Станок фрезерный 6Р33Г	станко-ч	38 740	36 660	1,1
7	Слесарь-сборщик	нормо-ч	21 000		
8	Наладчик	нормо-ч	9 080		
9	Контролер	нормо-ч	3 370		
10	Слесарь-ремонтник	нормо-ч	2 706		

Рис. 3.13. Результат расчета мощности

Два последних столбца таблицы *Баланс мощности* рассчитываются только по тем строкам, в которых находится оборудование (строки 3–6). Вектор *НМ* (планового наличия мощности) – это итоговый плановый фонд времени в часах, рассчитанный на листе *ФондВрСтан* в последнем столбце. Следовательно, для отображения вектора *НМ* в таблице *Баланс мощности* нужно воспользоваться ссылкой на соответствующую ячейку листа *ФондВрСтан*. Последний столбец таблицы *Баланс мощности* вычисляется по следующей формуле:

$$\text{Отношение потребности к наличию} = \text{ПМ} / \text{НМ}.$$

Балансировка мощности

Балансировка потребности в мощности занимает центральное место в планировании производственно-коммерческой деятельности предприятия. С одной стороны, имеющиеся ресурсы должны быть использованы, иначе не окупятся затраты на их приобретение и содержание, с другой стороны, если для выполнения производственной программы требуется ресурсов больше, чем их имеется в наличии, план нереален и выполнен не будет.

Часто ограничение на потребность в ресурсах записывается в виде неравенств следующего вида:

$$A * x \leq b, x \geq 0,$$

где A – матрица удельных норм расхода ресурсов;

x – объем производства продукции;

b – вектор планового наличия ресурсов.

Другими словами, потребность в ресурсах на план производства не должна превышать их планового наличия. Такое определение области производственных возможностей представляется излишне строгим. Более разумно требовать, чтобы потребности примерно соответствовали наличию следующих выражений:

$$A * x \approx b, x \geq 0.$$

Дело в том, что на практике ограничения на наличие ресурсов не являются абсолютно жесткими. Ресурсы можно разделить, по крайней мере, на четыре категории по степени жесткости (мягкости) ограничений на их наличие:

1. Наиболее мягкими являются ограничения на наличие материалов и комплектующих. Это ограничение было ключевым в плановой экономике, когда материальные ресурсы распределялись централизованно (фондирование). В рыночной экономике можно купить почти все, что нужно, и практически в любом количестве (чем больше, тем лучше для продавца). Поэтому при разработке плана производства ограничения на материалы и комплектующие нужно учитывать только в редчайших и очень специфичных случаях.

2. Более жестким, но все еще весьма мягким, является ограничение на наличие трудовых ресурсов (в разрезе должностей и профессий). Нынешняя ситуация на рынке труда такова, что нанять дополнительных работников не составляет особого затруднения. Многое здесь зависит, конечно, от требований к квалификации работника. Найти (или обучить) электросварщика значительно труднее, чем найти подсобного рабочего. Аналогичным образом потерять квалифицированного специалиста значительно тяжелее, чем сократить неквалифицированного работника. Поэтому, чем выше требования к квалификации персонала, тем более жестким является ограничение на потребность в трудовых ресурсах.

3. Еще жестче, чем предыдущее, ограничение на потребность в оборотном капитале. В принципе, дефицит оборотного капитала может быть покрыт за счет банковского кредита, эмиссии ценных бумаг, привлечения партнеров и товарного кредита у поставщиков.

4. Наиболее жестким является ограничение на потребность в станко-часах работы оборудования. Оборудование не может приобретаться и продаваться так же быстро и просто, как материалы и комплектую-

шие. Состав и структура станочного парка определяются в долгосрочном плане и на период, не меньший, чем средний период амортизации. Основные производственные фонды – наименее подвижная часть капитала. Однако даже это ограничение не является абсолютно жестким. Во-первых, в некоторых пределах можно варьировать коэффициент сменности. Во-вторых, в случае перегрузки оборудования возможна сверхурочная работа. Поэтому балансировку мощности можно считать удовлетворительной, если потребность в мощности отклоняется от планового наличия, пределы допустимого отклонения определяются конкретными условиями производства. В нашем примере отклонение составляет от –10% до +20% (см. столбец Е рис. 3.13).

Балансировка мощности проводится следующим образом. Если потребность существенно отличается от наличия, надо просто ввести новый план производства. Никаких новых формул вводить не надо: пересчет потребности произойдет автоматически после ввода нового плана производства. Например, если увеличить план выпуска изделия АТ-1000 на 10%, введя в ячейку С4 формулу =110000 * 0,9, получится новый баланс мощности, приведенный на рис. 3.14, в котором отклонение уже составит от –10% до +10%. Дальнейшие расчеты будут проведены с исходным значением план выпуска изделия АТ-1000, т. е. равным 110 000 штук.

	A	B	C	D	E
1	Баланс мощности				
2	Наименование группы оборудования или профессии	Ед. изм.	Потребность в мощности	Наличие мощности	Отношение потребности к наличию
3	Станок токарно-винторезный РТ2505	станко-ч	20 340	17 811	1,1
4	ПрессП-511Б	станко-ч	25 722	29 578	0,9
5	Станок фрезерный 6610	станко-ч	10 044	9 165	1,1
6	Станок фрезерный 6Р83Г	станко-ч	37 530	36 660	1,0
7	Слесарь-сборщик	нормо-ч	20 340		
8	Наладчик	нормо-ч	8 772		
9	Контролер	нормо-ч	3 249		
10	Слесарь-ремонтник	нормо-ч	2 706		

Рис. 3.14. Результат расчета мощности после балансировки

3.5. Планирование потребности в материалах

Расчет потребности в материалах нужен для разработки детального плана материально-технического снабжения (выбор способа поставок – транзитные или складские, установление размера партии закупок и т. д.), расчета плановой себестоимости и финансового плана. Различают валовую и чистую потребность в материалах.

Под валовой потребностью понимается потребность в материалах без учета запасов в начале и в конце планово-учетного периода. Валовая потребность в материалах вычисляется на основе норм расхода материалов по изделиям и проекта годовой производственной программы. Валовая потребность в материале рассчитывается по формуле

$$B_i = \sum_j нрм_{ij} * x_j,$$

где i – наименование (номер, шифр, код) материала;
 j – наименование (номер, шифр, код) изделия;
 B_i – валовая потребность в материале i -го наименования;
 $нрм_{ij}$ – норма расхода материала i на одно изделие j ;
 x_j – план производства изделия j .

То же самое уравнение валовой потребности в материалах имеет следующий вид в матричной форме:

$$B = нрм * x,$$

где B – вектор валовой потребности в материалах;
 $нрм$ – матрица норм расхода материала;
 x – вектор объема производства продукции (лист *Произв*).

Для расчета вектора B лучше всего воспользоваться функцией МУМНОЖ.

Чистая потребность – это тот объем материала, который надо закупить в планово-учетном периоде. Вычисляется по формуле

$$Ч = B - Ост_n + Ост_k,$$

где $Ч$ – чистая потребность в материале;

$В$ – валовая потребность в материале;

$Ост_n, Ост_k$ – остатки (запас) материала на начало и конец планово-учетного периода, соответственно.

Среднегодовая величина остатков материала ($Ост_{cp}$) определяется выражением

$$Ост_{cp} = (Ост_n + Ост_k) / 2.$$

Среднегодовая стоимость запаса материала ($Ст_{cp}$) вычисляется по формуле

$$Ст_{cp} = Ост_{cp} * ЦМ,$$

где $ЦМ$ – цена материала.

Нормы расхода материалов и цены на них находятся на листе *НорМат* (рис. 3.15). Расчет потребности в материалах реализован на листе *Матер*. Структура расчетной таблицы приведена на рис. 3.16. Результат вычислений показан на рис. 3.17.

	А	В	С	Д	Е
1	Нормы расхода материалов и цены на них				
2	Материал / комплектующее		Изделие		Цена материала
3	Наименование	Ед. изм.	АК-3	АТ-1000	
4	Лист стальной	кг	1,5	1,8	1 920
5	Уголки	кг	9	7	2 960
6	Лак	кг	6	6	1 280
7	Краска	кг	3,5	3,5	720
8	Электрооборудование	комплект	1	1	60 000
9	Цветные металлы	кг	0,4	0,6	2 080

Рис. 3.15. Структура данных листа *НорМат*

	А	В	С	Д	Е	Ф	Г	Н
1	Расчет потребности в материалах							
2	Материалы/комплектующие		Остатки		Потребность		Среднегодо- вые остатки	Среднегодо- вая стои- мость
3	Наименование	Ед. изм.	на начало периода	на конец периода	валовая	чистая		
4	Лист стальной	кг	435	415				
5	Уголки	кг	18 090	18 090				
6	Лак	кг	12 400	12 600				
7	Краска	кг	6 040	6 044				
8	Электрооборудование	комплект	2 000	1 960				
9	Цветной металл	кг	800	800				
10							Всего	

Рис. 3.16. Структура данных листа *Матер*

	А	В	С	Д	Е	Ф	Г	Н
1	Расчет потребности в материалах							
2	Материалы/комплектующие		Остатки		Потребность		Среднегодо- вые остатки	Среднегодо- вая стои- мость
3	Наименование	Ед. изм.	на начало периода	на конец периода	валовая	чистая		
4	Лист стальной	кг	435	415	558 000	557 980	425	816 000
5	Уголки	кг	18 090	18 090	2 930 000	2 930 000	18 090	53 546 400
6	Лак	кг	12 400	12 600	2 100 000	2 100 200	12 500	16 000 000
7	Краска	кг	6 040	6 044	1 225 000	1 225 004	6 042	4 350 240
8	Электрооборудование	комплект	2 000	1 960	350 000	349 960	1 980	118 800 000
9	Цветной металл	кг	800	800	162 000	162 000	800	1 664 000
10							Всего	195 176 640

Рис. 3.17. Результат расчета потребности в материалах

3.6. Планирование численности персонала

Промышленно-производственный персонал подразделяют на следующие группы: рабочие, инженерно-технические работники, служащие, младший обслуживающий персонал, ученики и работники охраны.

Рабочие подразделяются на основных и вспомогательных. К основным производственным рабочим относят тех рабочих, которые непосредственно воздействуют на предметы труда. Вспомогательные рабочие – это рабочие, обеспечивающие нормальный ход производственного процесса (ремонтные, транспортные, складские работники и т. п.).

Планирование численности основных производственных рабочих осуществляется в два этапа. На первом этапе определяется потребность в численности основных производственных рабочих, на втором в ре-

зультате сравнения со списочной численностью устанавливается плановая численность основных производственных рабочих.

Целесообразно различать рабочих-станочников и рабочих на ручных операциях. В рассматриваемом здесь упрощенном примере в производстве заняты станочники следующих профессий: токари, штамповщики и фрезеровщики (причем фрезеровщики, занятые в различных группах фрезерных станков, учитываются отдельно). На ручных операциях используются рабочие следующих профессий: слесари-сборщики, наладчики (готовой продукции, а не оборудования) и контролеры, осуществляющие технический контроль.

Для станочников действительный фонд времени одного рабочего данной профессии связан с действительным фондом времени одного станка. Потребность в их численности определяется по формуле

$$Чс = ПМ * K_{см} / Дч,$$

где $Чс$ – потребность в численности рабочих-станочников;

$ПМ$ – потребность в мощности рабочих-станочников (лист *Мощн*);

$K_{см}$ – количество смен (лист *ФондВрСтан*);

$Дч$ – действительный фонд времени рабочего на единице оборудования в часах (лист *ФондВрСтан*).

Для рабочих на ручных операциях действительный фонд времени одного рабочего данной профессии определяется при расчете баланса рабочего времени. Потребность в их численности определяется по формуле

$$Чр = ПМ / Дч,$$

где $Чр$ – потребность в численности рабочих на ручных операциях;

$ПМ$ – потребность в мощности рабочих на ручных операциях (лист *Мощность*);

$Дч$ – действительный фонд рабочего времени в часах (лист *ФондВрРучн*).

Расчетная таблица для планирования численности персонала находится на листе *Числ* и была приведена на рис. 3.8.

В результате деления на значение $Дч$, как правило, получаются дробные числа. Поэтому далее расчетная численность преобразуется в реальную плановую численность путем округления до ближайшего большего целого числа. Для этой цели можно использовать функцию Excel ОКРУГЛВВЕРХ. Результаты расчета численности основных производственных рабочих показаны на рис. 3.18.

	A	B	C	D	E	F	G
1	План по труду						
2	Основные производственные рабочие			Вспомогательные рабочие		ИТР и служащие	
3	Наименование профессии	Численность расчетная	Численность плановая	Наименование профессии	Численность плановая	Наименование должности	Численность плановая
4	Токарь	11,55	12	Слесарь-ремонтник	2	Директор	1
5	Штамповщик	14,38	15	Электрик	3	Зам. директора по сбыту	1
6	Фрезеровщик	5,70	6	Кладовщица	2	Гл. инженер	1
7	Фрезеровщик	21,13	22	Уборщица	4	Гл. бухгалтер	1
	Слесарь-сборщик	12,35	13	Грузчик	1	Менеджер	7
8	Наладчик	5,34	6	Водитель	3	Мастер	7
9	Контролер	1,93	2	Водитель автокара	1	Бухгалтер	2
10	Итого сдельщики		76	Итого повременщики	16	Итого ИТР и служащие	20
11	Численность списочная всего			112			

Рис. 3.18. Результат расчета численного состава

Численность вспомогательных производственных рабочих, ИТР и служащих может задаваться, как исходные данные, или рассчитываться. Так, например, численность ремонтников определяется исходя из трудоемкости ремонтных работ и эффективного (действительного) фонда времени одного рабочего-ремонтника по той же формуле, как и для рабочих на ручных операциях:

$$Чрем = ПМ / Дч,$$

где $Чрем$ – потребность в численности рабочих-ремонтников;

$ПМ$ – потребность в мощности рабочих-ремонтников (лист *Произв*);

$Дч$ – действительный фонд рабочего времени в часах (лист *ФондВрРучн*).

Значение $Чрем$ также может быть дробным. Однако на листе *Числ* отдельного столбца для расчетной численности не предусмотрено. Поэтому формула для нахождения планового значения $Чрем$ должна находиться внутри функции ОКРУГЛВВЕРХ.

Численность остальных вспомогательных производственных рабочих, ИТР и служащих сразу задана в расчетной таблице. Внизу каждой группы рассчитывается ее численность с помощью функции СУММ, а еще ниже – общая списочная численность персонала.

Окончательный итог расчета численности сотрудников приведен на рис. 3.18. Автоматически при расчете численности сотрудников обновляется лист *ФондВрРучн*, где определяется итоговый плановый фонд времени рабочих на ручных операциях. Новое содержимое данного листа приведено на рис. 3.19.

	А	В	С	Д	Е	Ф
1	Календарный фонд	Выходные и праздники	Номинальный фонд рабочего времени, дни	Номинальная продолжительность смены, ч	Льготное сокращение смены	Плановая продолжительность смены, ч
2	365	109	256	8	0,2	7,8
3	Плановые неявки на работу	Очередные отпуска	Дополнительные отпуска	Отпуска по временной нетрудоспособности	Действительный фонд рабочего времени, дни	Действительный фонд рабочего времени, ч
4	4	30	0	4	218	1 700,4
5	Расчет планового фонда времени рабочих на ручных операциях					
6	Наименование профессии	Численность	Итоговый плановый фонд			
7	Слесарь-сборщик	13	22 105,2			
8	Наладчик	6	10 202,4			
9	Контролер	2	3 400,8			
10	Слесарь-ремонтник	2	3 400,8			
11	Электрик	3	5 101,2			
12	Кладовщик	2	3 400,8			
13	Уборщица	4	6 801,6			
14	Грузчик	1	1 700,4			
15	Водитель	3	5 101,2			
16	Водитель автокара	1	1 700,4			

Рис. 3.19. Результат расчета планового фонда времени рабочих на ручных операциях

3.7. Планирование фонда оплаты труда

Существуют две основные формы оплаты труда: сдельная и повременная. Сдельно обычно оплачивается труд основных производственных рабочих, повременной – труд вспомогательных рабочих и основных производственных рабочих в непрерывном производстве. Оклады ИТР и служащих – по сути дела та же повременная оплата труда. Отличие только в том, что ИТР и служащим устанавливается месячный оклад и их месячный заработок не зависит от числа рабочих часов в месяце, а месячный заработок рабочих зависит от часовой тарифной ставки и числа рабочих часов в месяце.

В чистом виде сдельная и повременная формы оплаты труда используются крайне редко. Обычно они совмещаются с системой премирования и на практике применяются сдельно-премиальная и сдельно-повременная формы оплаты труда. Важной задачей планирования фонда заработной платы (ФЗП) является установление показателей, на основании которых начисляются и выплачиваются премии. Общий подход к решению таков, что премии рабочим устанавливаются в зависимости от выполнения плана реализации продукции (или непосредственно от объема реализации), ИТР и служащим – в зависимости от выполнения плана по прибыли. Помимо основной зарплаты, работникам предприятия устанавливаются доплаты и надбавки к основной зарплате.

Фонд прямой сдельной зарплаты рабочих-сдельщиков рассчитывается исходя из расценок за изделие и плана выпуска изделий по формуле

$$\Phi ЗП_{сд} = \text{Тариф} * ПМ,$$

где $\Phi ЗП_{сд}$ – фонд заработной платы рабочих-сдельщиков на планово-учетный период;

Тариф – часовая тарифная ставка;

ПМ – потребность в мощности соответствующего оборудования (лист *Мощность*).

Расчет фонда заработной платы реализован на листе *ФЗП*. Часовые тарифные ставки расположены на листе *Тариф* (рис. 3.20). Структура расчетной таблицы приведена на рис. 3.21. Результат вычислений показан на рис. 3.22.

	A	B	C	D	E
1	Часовые тарифные ставки и оклады				
2	Рабочие		ИТР и служащие		
3	Наименование профессии	Часовая тарифная ставка (руб./ч)	Наименование должности	Базовый коэффициент	Оклад (руб./мес.)
4	Токарь	2 000	Директор		1 200 000
5	Штамповщик	1 680	Зам. директора по сбыту	0,85	
6	Фрезеровщик	1 840	Гл. инженер	0,8	
7	Фрезеровщик	1 840	Гл. бухгалтер	0,8	
8	Слесарь-сборщик	2 160	Менеджер	0,4	
9	Наладчик	2 800	Мастер	0,35	
10	Контролер	720	Бухгалтер	0,3	
11	Слесарь-ремонтник	2 080			
12	Электрик	1 920			
13	Кладовщик	880			
14	Уборщица	400			
15	Грузчик	960			
16	Водитель	2 560			
17	Водитель автокара	1 280			

Рис. 3.20. Структура данных листа *Тариф*

	A	B	C	D	E	
1	Фонд заработной платы					
2	Основные производственные рабочие (сдельщики)		Вспомогательные рабочие (повременщики)		ИТР и служащие	
3	Наименование профессии	Фонд зарплаты	Наименование профессии	Фонд зарплаты	Наименование должности	Фонд зарплаты
4	Токарь		Слесарь-ремонтник		Директор	
5	Штамповщик		Электрик		Зам. директора по сбыту	
6	Фрезеровщик		Кладовщик		Гл. инженер	
7	Фрезеровщик		Уборщица		Гл. бухгалтер	
8	Слесарь-сборщик		Грузчик		Менеджер	
9	Наладчик		Водитель		Мастер	
10	Контролер		Водитель автокара		Бухгалтер	
11	Итого сдельщики		Итого повременщики		Итого ИТР и служащие	
12	Структура ФЗП, %					
13	Общий ФЗП				Среднемесячный ФЗП	

Рис. 3.21. Структура данных листа *ФЗП*

	A	B	C	D	E	F
1	Фонд заработной платы					
2	Основные производственные рабочие		Вспомогательные рабочие		ИТР и служащие	
3	Наименование профессии	Фонд зарплаты	Наименование профессии	Фонд зарплаты	Наименование должности	Фонд зарплаты
4	Токарь	42 000 000	Слесарь-ремонтник	7 073 664	Директор	14 400 000
5	Штамповщик	44 654 400	Электрик	9 794 304	Зам. директора по сбыту	12 240 000
6	Фрезеровщик	19 209 600	Кладовщик	2 992 704	Гл. инженер	11 520 000
7	Фрезеровщик	71 281 600	Уборщица	2 720 640	Гл. бухгалтер	11 520 000
8	Слесарь-сборщик	45 360 000	Грузчик	1 632 384	Менеджер	40 320 000
9	Наладчик	25 424 000	Водитель	13 059 072	Мастер	35 280 000
10	Контролер	2 426 400	Водитель автокара	2 176 512	Бухгалтер	8 640 000
11	Итого сдельщики	250 356 000	Итого повременщики	39 449 280	Итого ИТР и служащие	133 920 000
12	Структура ФЗП, %					
13	Общий ФЗП		423 725 280		Среднемесячный ФЗП	
					35 310 440	

Рис. 3.22. Результат расчета фонда заработной платы

Схема расчета ФЗП рабочих-повременщиков в основном та же, что и расчета фонда оплаты труда рабочих-сдельщиков. Главное отличие в том, что вместо потребности в мощности рабочих-сдельщиков используется итоговый плановый фонд времени рабочих-повременщиков, рассчитываемый по формуле

$$\text{ФЗП}_{\text{вр}} = \text{Тариф} * \text{ИПФ},$$

где $\text{ФЗП}_{\text{вр}}$ – фонд оплаты труда рабочих-повременщиков на планово-учетный период;

Тариф – часовая тарифная ставка рабочих-повременщиков;

ИПФ – итоговый плановый фонд времени рабочих-повременщиков (лист *ФондВрРучн*).

ФЗП руководителей, ИТР и служащих рассчитывается исходя из штатного расписания, в котором указана численность работников этих категорий, и должностных окладов. Для расчета окладов наиболее часто используется метод базовых коэффициентов: базовый коэффициент директора принимается за 1 и оклад работника на любой другой должности приравнивается к окладу директора, умноженному на базовый коэффициент данной должности. Формула для этих расчетов имеет следующий вид:

*Оклад сотрудника = Базовый коэффициент * Оклад директора.*

Расчетная таблица для окладов сотрудников находится на листе *Тариф* (см. рис. 3.20). Результат вычислений приведен на рис. 3.23.

Часовые тарифные ставки и оклады				
Рабочие		ИТР и служащие		
Наименование профессии	Часовая тарифная ставка (руб./ч)	Наименование должности	Базовый коэффициент	Оклад (руб./мес.)
Токарь	2 000	Директор		1 200 000
Штамповщик	1 680	Зам. директора по сбыту	0,85	1 020 000
Фрезеровщик	1 840	Гл. инженер	0,8	960 000
Фрезеровщик	1 840	Гл. бухгалтер	0,8	960 000
Слесарь-сборщик	2 160	Менеджер	0,4	480 000
Наладчик	2 800	Мастер	0,35	420 000
Контролер	720	Бухгалтер	0,3	360 000
Слесарь-ремонтник	2 080			
Электрик	1 920			
Кладовщик	880			
Уборщица	400			
Грузчик	960			
Водитель	2 560			
Водитель автокара	1 280			

Рис. 3.23. Результат расчета окладов сотрудников

ФЗП руководителей, ИТР и служащих рассчитывается по следующей формуле:

$$ФЗП_{итр} = 12 * Численность * Оклад,$$

где $ФЗП_{итр}$ – фонд оплаты труда руководителей, ИТР и служащих;

Численность – численность руководителей, ИТР и служащих (лист *Числ*);

Оклад – должностной оклад (лист *Тариф*).

Результат расчета ФЗП руководителей, ИТР и служащих показан на рис. 3.22.

Итоговые показатели по ФЗП (общий годовой ФЗП, среднемесячный ФЗП, структура ФЗП) рассчитаны в нижней части таблицы *Фонд заработной платы* (см. рис. 3.22) по следующим формулам:

$$\begin{aligned} \text{Общий ФЗП} = & \text{Итого сдельщики} + \text{Итого повременщики} + \\ & + \text{Итого ИТР и служащие}; \end{aligned}$$

$$\text{Структура ФЗП} = \text{Итого по группе} / \text{Общий ФЗП};$$

$$\text{Среднемесячный ФЗП} = \text{Общий ФЗП} / 12.$$

Результат расчетов общих показателей по ФЗП показан на рис. 3.22.

3.8. Планирование себестоимости товарной продукции

Основные цели планирования себестоимости продукции – это определение необходимой суммы затрат на производство и реализацию продукции и выявление резервов снижения издержек производства. Отнесение тех или иных затрат на себестоимость продукции регламентируется правительством.

Особенно большое значение для планирования деятельности предприятия имеет разделение затрат на переменные и постоянные. Постоянные затраты не изменяются при изменении объемов производства и продаж в довольно широких пределах, переменные затраты зависят от объемов производства и продаж. Отнесение тех или иных затрат к постоянным или переменным зависит от конкретных условий и организации производства и реализации продукции. Так, зарплата менеджеров по продажам (торговых агентов) относится к постоянным затратам, если они получают только оклады, и к переменным, если они получают только процент от объема продаж. Обычно, к переменным затратам относятся следующие:

- прямые материальные затраты (стоимость материалов и покупных комплектующих, из которых изготовлено изделие, затраты на электроэнергию, сжатый воздух, пар и воду, если они учитываются не «котловым методом», а с помощью специальных счетчиков);

- прямая сдельная заработная плата;
- премии, зависящие от объемов производства и продаж.

К постоянным затратам чаще всего относятся следующие:

- затраты на материалы, топливо, энергию, воду, не используемые непосредственно в технологических процессах;

- амортизация;
- зарплата ИТР, служащих, руководителей и вспомогательных рабочих;
- прочие общепроизводственные и общехозяйственные расходы.

Расчет амортизации

Для построения годового плана предприятия надо знать общую сумму годовой амортизации, независимо от конкретной техники начисления текущей амортизации. Поэтому можно ограничиться укрупненным расчетом. Исходные данные для этого расчета расположены на листе Excel *ОПФ*, структура которого приведена на рис. 3.24. Значения расчетных показателей определяются на этом же листе по следующим формулам:

$$A = BC * \text{Норма};$$

$$ОС_{кон} = ОС_{нач} - A,$$

где A – годовая сумма амортизации по группе оборудования;

BC – балансовая стоимость единицы оборудования из группы;

$Норма$ – норма амортизации единицы оборудования из группы;

$ОС_{кон}$ – остаточная стоимость группы оборудования на конец года;

$ОС_{нач}$ – остаточная стоимость группы оборудования на начало года.

В строке *Всего* необходимо отразить итоговые результаты показателей с учетом количества единиц оборудования в каждой группе. Необходимое значение представляет собой скалярное произведение вектора показателя на вектор количества единиц оборудования, т. е. определяется по формулам:

$$\text{Всего } BC = BC * \text{кол};$$

$$\text{Всего } ОС_{нач} = ОС_{нач} * \text{кол};$$

$$\text{Всего } ОС_{кон} = ОС_{кон} * \text{кол};$$

$$\text{Всего } A = A * \text{кол},$$

где BC , $ОС_{нач}$, $ОС_{кон}$, A , $кол$ – соответственно, векторы балансовой стоимости, остаточной стоимости на начало года, остаточной стоимости на конец года, амортизации, количества единиц оборудования.

Все векторы расположены в столбцах. Поэтому для нахождения скалярного произведения можно воспользоваться функцией СУММПРОИЗВ, аргументами которой являются векторы. Результат амортизационных расчетов показан на рис. 3.25.

	A	B	C	D	E	F	G
1	Основные производственные фонды						
2	Наименование группы оборудования	Балансовая стоимость	Норма амортизации	Остаточная стоимость на начало года	Остаточная стоимость на конец года	Количество единиц оборудования в группе	Амортизация единицы оборудования
3	единицы оборудования						
4	Станок токарно-винторезный РТ2505	5 360 000	0,25	31 000		7	
5	Пресс П-511Б	10 400 000	0,25	45 700		16	
6	Станок фрезерный 6610	6 480 000	0,25	74 200		5	
7	Станок фрезерный 6Р83Г	5 120 000	0,25	25 500		20	
8	Корпус 1	213 920 000	0,05	267 400		1	
9	Корпус 2	58 400 000	0,1	438 000		1	
10	Прочие	20 960 000	0,25	131 000		12	
11	Всего						

Рис. 3.24. Структура данных листа *ОПФ*

	A	B	C	D	E	F	G
1	Основные производственные фонды						
2	Наименование группы оборудования	Балансовая стоимость	Норма амортизации	Остаточная стоимость на начало года	Остаточная стоимость на конец года	Количество единиц оборудования в группе	Амортизация единицы оборудования
3	единицы оборудования						
4	Станок токарно-винторезный РТ2505	5 360 000	0,25	31 000	2 480 000	7	1 340 000
5	Пресс П-511Б	10 400 000	0,25	45 700	3 656 000	16	2 600 000
6	Станок фрезерный 6610	6 480 000	0,25	74 200	5 936 000	5	1 620 000
7	Станок фрезерный 6Р83Г	5 120 000	0,25	25 500	20 400 000	20	1 280 000
8	Корпус 1	213 920 000	0,05	267 400	21 392 000	1	10 696 000
9	Корпус 2	58 400 000	0,1	438 000	35 040 000	1	5 840 000
10	Прочие	20 960 000	0,25	131 000	10 480 000	12	5 240 000
11	Всего	862 560 000		695 728 000	695 728 000		164 096 000

Рис. 3.25. Результат расчета амортизации

Целесообразно постоянные и переменные затраты планировать отдельно по экономическим показателям. Структура одного из возможных способов представления информации на листе *Себест* приведена на рис. 3.26.

	А	В
1	Расчет плановой себестоимости товарной продукции	
2	Показатель	Сумма (тыс. руб.)
3	Постоянные затраты	
4	Амортизация	
5	Зарплата с отчислениями во внебюджетные фонды	
6	Прочие	
7	Переменные затраты	
8	Зарплата с отчислениями во внебюджетные фонды	
9	Материалы	
10	Прочие	
11	ИТОГО	

Рис. 3.26. Структура данных листа *Себест*

Расчет показателей затрат на план выпуска производится по следующим формулам:

$$\text{Постоянные затраты} = \text{Амортизация} + \text{ФЗП}_{\text{пост}} + \text{Прочие постоянные};$$

$$\text{Амортизация} = \text{Всего } A / 1000;$$

$$\text{ФЗП}_{\text{пост}} = (1 + \alpha) * (\text{Итого повременщики} + \text{Итого ИТР и служащие}) / 1000;$$

$$\text{Прочие постоянные} = 0,09 * \text{ФЗП}_{\text{пост}};$$

$$\text{Переменные затраты} = \text{ФЗП}_{\text{пер}} + \text{Материалы} + \text{Прочие переменные};$$

$$\text{ФЗП}_{\text{пер}} = (1 + \alpha) * \text{Итого сдельщики} / 1000;$$

$$\text{Материалы} = (\text{ЦМ}^T * \text{нрм}) * x^T / 1000;$$

$$\text{Прочие переменные} = 0,15 * (\text{ФЗП}_{\text{пер}} + \text{Материалы});$$

$$\text{ИТОГО} = \text{Постоянные затраты} + \text{Переменные затраты},$$

где *Всего А* – итоговое значение амортизация за год (лист *ОПФ*);
ФЗП_{пост} – зарплата постоянная с отчислениями во внебюджетные фонды;
 α – ставка единого социального налога (лист *КритОбъем*);
Итого повременщики – итоговое значение ФЗП рабочих-повременщиков (лист *ФЗП*);
Итого ИТР и служащие – итоговое значение ФЗП ИТР и служащих (лист *ФЗП*);
ФЗП_{пер} – зарплата переменная с отчислениями во внебюджетные фонды;
Итого сдельщики – итоговое значение ФЗП рабочих-сдельщиков (лист *ФЗП*);
 ЦМ^T – транспонированный вектор цен на материалы (лист *ЦеныМат*);
нрм – матрица норм расхода материала (лист *НорМат*);
 x^T – транспонированный вектор объема производства продукции (лист *Произв*).

Расчет себестоимости материалов должен быть реализован с помощью формулы массивов. Вектор цен на материалы первоначально транспонируется функцией ТРАНСП (преобразуется из столбца в строку), а затем функцией МУМНОЖ умножается на матрицу норм расхода материала. Промежуточным результатом матричного произведения является вектор-строка, которую далее нужно скалярно умножить на вектор-строку объема производства продукции. Однако объем производства продукции задан в столбце. Поэтому он первоначально транспонируется. Далее можно использовать функцию СУММПРОИЗВ.

Данные на листе *Себест* делятся на 1000. Дело в том, что соответствующие показатели на других листах рассчитаны в рублях, а на листе *Себест* они должны быть приведены в тысячах рублей. Прочие затраты устанавливаются на основе фактических данных за предплановый (базовый) период. В примере предполагается, что прочие затраты в составе постоянных затрат равны 9% от постоянной части ФЗП с отчислениями во внебюджетные фонды, прочие затраты в составе переменных затрат равны 15% от суммы затрат на материалы и переменной части ФЗП с отчислениями во внебюджетные фонды. Себестоимость товарного выпуска рассчитывается как сумма постоянных и переменных затрат.

Результат расчета постоянных и переменных затрат приведен на рис. 3.27.

А	В
1	Расчет плановой себестоимости товарной продукции
2	Показатель Сумма (тыс. руб.)
3	Постоянные затраты 422 043
4	Амортизация 164 096
5	Зарплата с отчислениями во внебюджетные фонды 236 649
6	Прочие 21 298
7	Переменные затраты 40 241 784
8	Зарплата с отчислениями во внебюджетные фонды 341 736
9	Материалы 34 651 120
10	Прочие 5 248 928
11	ИТОГО 40 663 828

Рис. 3.27. Результат расчета себестоимости

3.9. Планирование прибыли и рентабельности

Главным итоговым показателем деятельности предприятия является прибыль, которая в нормативных документах по налогообложению часто именуется финансовым результатом. Плановая величина прибыли рассчитывается как разность между объемом реализации (без учета НДС) и себестоимостью реализуемой продукции по формуле

$$ПП = Реал - СБ,$$

где *ПП* – плановая величина прибыли от реализации продукции;

Реал – объем реализации (без учета НДС);

СБ – себестоимость реализуемой продукции (лист *Себест*).

Объем реализации при равномерном объеме продаж определяется по формуле

$$Реал = ЦП * x / 1000,$$

где *ЦП* – вектор цен единицы продукции (лист *Произв*);

x – вектор планового объема производства продукции (лист *Произв*).

Рентабельность продукции – это отношение плановой прибыли к себестоимости продукции, а *рентабельность капитала* – отношение прибыли к сумме основного и оборотного капиталов. Рентабельность продукции и рентабельность капитала вычисляется по формулам:

$$РП = ПП / СБ;$$

$$РК = ПП / (ОснК + ОборК),$$

где *РП* – рентабельность продукции;

РК – рентабельность капитала;

ОснК – величина основного капитала, т. е. итоговая балансовая стоимость основных фондов (лист *ОПФ*), которую нужно перевести в тысячи рублей, разделив на 1000;

ОборК – величина оборотного капитала, задаваемая в расчетной таблице.

Расчетная таблица для планирования прибыли и рентабельности находится на листе *Прибыль* (рис. 3.28). Результат вычислений приведен на рис. 3.29.

	А	В	С
1	Расчет прибыли и рентабельности		
2	Наименование показателя	Ед. изм.	Значение
3	Объем реализации	тыс. руб.	
4	Себестоимость	тыс. руб.	
5	Плановая прибыль	тыс. руб.	
6	Основной капитал	тыс. руб.	7 318 160
7	Оборотный капитал	тыс. руб.	
8	Рентабельность капитала	%	
9	Рентабельность продукции	%	

Рис. 3.28. Структура данных листа *Прибыль*

А	В	С
1	Расчет прибыли и рентабельности	
2	Наименование показателя	Ед. изм. Значение
3	Объем реализации	тыс. руб. 47 944 000
4	Себестоимость	тыс. руб. 40 663 828
5	Плановая прибыль	тыс. руб. 7 280 172
6	Основной капитал	тыс. руб. 862 560
7	Оборотный капитал	тыс. руб. 7 318 160
8	Рентабельность капитала	% 89.0%
9	Рентабельность продукции	% 17.9%

Рис. 3.29. Результат расчета прибыли и рентабельности

Раздел 4. ИМПОРТИРОВАНИЕ ДАННЫХ В MS EXCEL ИЗ ВНЕШНИХ ИСТОЧНИКОВ

4.1. Основные теоретические сведения

MS Query – приложение, позволяющее подключаться к внешним источникам данных (как правило, к базам данных) и импортировать данные в рабочую книгу MS Excel автоматически, без применения средств программирования.

Данное приложение можно также использовать и для чтения таблиц MS Excel (вместо команды *Файл* → *Открыть*) в следующих случаях:

- если файл данных слишком большой (*MS Query* размещает в оперативной памяти только те данные исходного файла, которые удовлетворяют заданным критериям);
- если с файлом данных одновременно работает несколько пользователей (подключение к файлу через *MS Query* позволяет обновить лист Excel при внесении изменений в базу данных другими пользователями);
- если нужно объединить несколько таблиц данных (*MS Query*, в отличие от Excel, позволяет объединять таблицы, которые имеют общие поля).

Установка надстройки *MS Query* выполняется с помощью цепочки команд:

C:/Program Files/Microsoft Office/Office/Library/ MSQuery/ файл Xlquery.xla.

Если MS Excel уже открыт, то нужно выполнить следующие команды:

Сервис → *Надстройки* → *Надстройка MS Query* → *ОК*.

4.2. Использование надстройки MS Query для импортирования в MS Excel данных, созданных в СУБД Access

4.2.1. Постановка задачи

В данном примере нужно выполнить следующее:

1. В MS Access создать базу данных с именем *Учет поставок.mdb*. Схема данных указанной базы данных представлена на рис. 4.1.
2. Заполнить таблицы базы данных в соответствии с рис. 4.2.
3. Создать в MS Query *новый источник данных* с именем *Поставки*. В качестве источника использовать базу данных, созданную в MS Access с именем *Учет поставок.mdb*.
4. Создать и сохранить в MS Query следующие запросы к созданному источнику данных:
 - список фактических поставок продукции, упорядоченный по дате поставки и наименованию продукции;
 - список фактических поставок продукции с наименованием «конфеты», упорядоченный по виду продукции и по дате;
 - список фактических поставок продукции, с наименованием «конфеты», количество которой превышает 15 единиц, упорядоченный по наименованию продукции и по дате;
 - список фактических поставок продукции в период с 12 марта по 15 марта 2005 г. (включительно);
 - расчет стоимости фактической поставки каждого вида продукции, упорядоченный по стоимости;
 - список продукции, стоимость фактической поставки которой превышает 100 000 р., упорядоченный по сумме фактической поставки;
 - список фактических поставок продукции по поставщику, упорядоченный по дате, наименование которого является параметром запроса;
 - список продукции, поставленной в период, границы которого являются параметрами запроса;
 - общее количество поставленной продукции каждого вида;
 - общая стоимость поставок продукции за каждую дату.
5. Выполнить созданные запросы, поместив результаты их выполнения на отдельные листы MS Excel в книгу с именем *Импортирование1(Поставки).xls*.
6. Отобразить в виде *сводной таблицы* общую стоимость поставок каждого вида продукции каждым поставщиком за каждую дату, предварительно подготовив данные для сводной таблицы в запросе MS Query. Сохранить сводную таблицу на рабочем листе с именем *СводнаяТаблица* в книге *Импортирование1(Поставки).xls*.

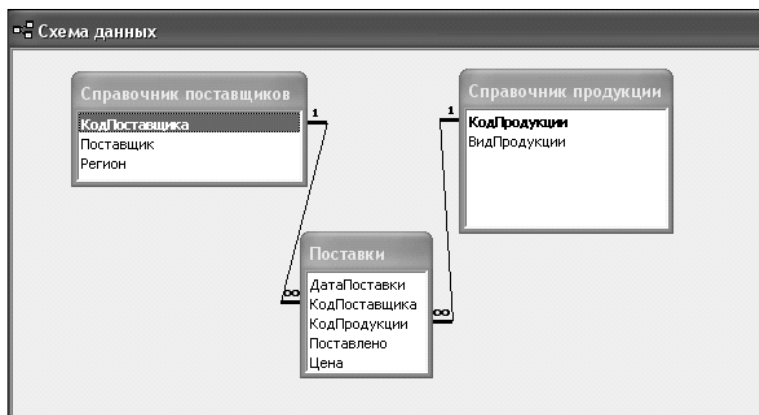


Рис. 4.1. Схема связей между таблицами в базе данных *Учет поставок*

а) **Справочник поставщиков : таблица**

	КодПоставщика	Поставщик	Регион
+	1	Тула	Центр
+	2	Симферополь	Крым
+	3	Мурманск	Север
+	4	Краснодар	Юг
+	5	Калининград	Запад
+	6	Хабаровск	Восток
+	7	Москва	Центр

б) **Справочник продукции : таблица**

	КодПродукции	ВидПродукции
+	1	чай
+	2	ноутбуки
+	3	кофе
+	4	компьютеры
+	5	мониторы
+	6	маршрутизаторы
+	7	консервы
+	8	овощи
+	9	конфеты
+	10	рыба

в) **Поставки : таблица**

	ДатаПоставки	КодПоставщик	КодПродукции	Поставлено	Цена
	12.03.2005	6	7	40	2 700р.
	12.03.2005	3	7	50	3 000р.
	13.03.2005	1	9	60	9 000р.
	13.03.2005	4	1	1000	3 500р.
	14.03.2005	2	8	100	5 000р.
	15.03.2005	7	6	5	120 000р.
	15.03.2005	7	5	3	350 000р.
	18.03.2005	5	3	120	5 000р.
	18.04.2005	3	10	150	7 000р.
	19.03.2006	5	9	20	12 000р.

Рис. 4.2. Состав таблиц базы данных *Учет поставок*

4.2.2. Создание в MS Query нового источника данных

Для создания нового источника данных в MS Query необходимо сделать следующее:

- выполнить команду *Данные → Внешние данные → Создать запрос* (для Excel 2002, Excel 2003 выполнить команду *Данные → Импорт внешних данных → Создать запрос*);
- в окне *Выбор источника данных* перейти на вкладку *База данных* и выбрать *Новый источник данных* → *OK*;
- в окне *Создание нового источника данных* необходимо указать имя источника (в нашей задаче имя источника – *Поставки*) и выбрать из раскрывающегося списка драйвер для доступа к базе данных данного типа (в нашем случае – *MS Access Driver*). Нажать кнопку *Связь* (рис. 4.3);
- в открывшемся окне нажать кнопку *Выбрать*;
- в следующем окне сначала выделить папку, в которой находится файл базы данных, затем из появившегося списка файлов баз данных указанного типа выделить имя нужного файла (*Учет поставок.mdb*);

и нажать *OK*, затем снова *OK*.

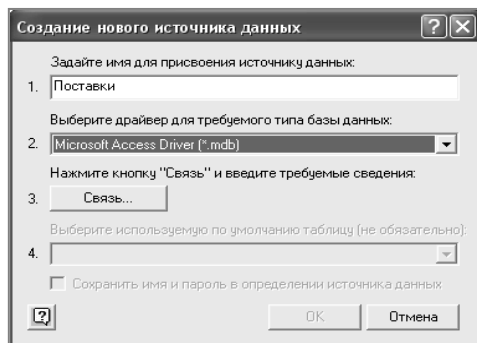


Рис. 4.3. Окно *Создание нового источника данных*

Если все выполнено правильно, то в окне *Выбор источника данных* появится имя созданного источника данных *Поставки*.

Итак, новый источник данных в MS Query подключен. Теперь окно *Выбор источника данных* можно закрыть.

Необходимо обратить внимание на то, что источник данных *Поставки* содержит в себе все таблицы, которые ранее находились в базе данных *Учет поставок.mdb*.

4.2.3. Создание и сохранение запросов в MS Query

Запросы в MS Query можно создавать двумя способами:

- с помощью *Мастера запросов* (для создания простых запросов);
- непосредственно в редакторе запросов MS Query (для создания сложных запросов и запросов с параметром).

Для создания сложных запросов необходимо снять флажок в поле *Использовать мастер запросов* окна *Выбор источника данных*.

Пример 1

Требуется создать с помощью *Мастера запросов* запрос 1 «Список фактически поставленной продукции, упорядоченный по дате и наименованию».

Решение

Для решения данного примера нужно произвести следующие действия:

- выполнить команду *Данные → Импорт внешних данных → Создать запрос*;
- в открывшемся окне *Выбор источника данных* на вкладке *База данных* выделить имя созданного источника данных (*Поставки*), убедиться, что флажок *Использовать мастер запросов* установлен, и нажать кнопку *ОК* (рис. 4.4);

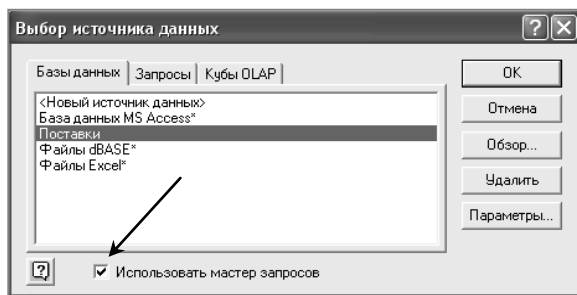


Рис. 4.4. Окно *Выбор источника данных*

- в открывшемся окне выбрать те поля из таблиц, которые участвуют в создании запроса. Для выбора полей воспользоваться подсказкой, представленной в виде табл. 4.1.

Таблица 4.1. Таблица выбора полей

Имя таблицы	Имя извлекаемого поля
СправочникПоставщиков	Поставщик
СправочникПродукции	ВидПродукции
Поставки	ДатаПоставки
	Поставлено
	Цена

При выборе полей нужно расположить их в следующем порядке:

ДатаПоставки,

ВидПродукции,

Поставщик,

Поставлено,

Цена;

- закончив выбор полей из таблиц и выполнив их упорядочение, нажать кнопку *Далее*;
- задать правило отбора данных (*ДатаПоставки* должна быть *не Null* (не пустой)), затем нажать кнопку *Далее*;
- указать следующий порядок сортировки: по *ДатаПоставки*, затем по *ВидПродукции*. Нажать кнопку *Далее*;
- для сохранения запроса нажать кнопку *Сохранить запрос*, ввести имя запроса *1 – фактические поставки* и нажать кнопку *Сохранить*. Запрос по умолчанию сохраняется в папку *Запросы* (C:/ Program Files/ Microsoft Office/) (рис. 4.5);

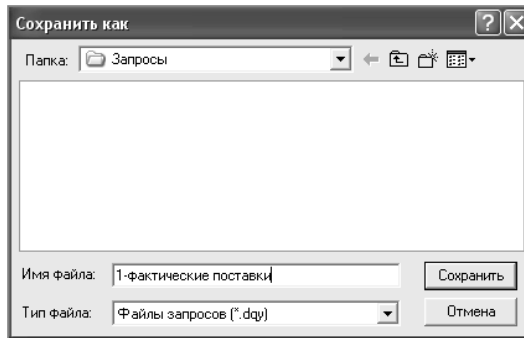


Рис. 4.5. Окно сохранения запроса в MS Query

- нажать кнопку *Отмена* в окне *Создание запроса*: заключительный шаг (т. е. отказаться от возврата данных в MS Excel).

Пример 2

Требуется создать с помощью *Мастера запросов* запрос 2 «Список фактических поставок продукции «конфеты», упорядоченный по виду продукции и по дате поставки».

Решение

Для решения данного примера нужно сделать следующее:

- выполнить команду *Данные → Импорт внешних данных → Создать запрос*;
- на вкладке *База данных* окна *Выбор источника данных* выделить имя созданного источника данных (*Поставки*) и, прежде, чем нажать *ОК*, проверить наличие флажка в поле *Использовать Мастер запросов*;
- в открывшемся окне нужно выбрать те поля из таблиц, которые участвуют в создании запроса. Для выбора полей необходимо снова воспользоваться подсказкой, представленной в виде табл. 4.1. Нажать кнопку *Далее*;
- задать правило отбора данных (поле *ВидПродукции* должно быть *равно*) и выбрать из вложенного списка слово *конфеты* (рис. 4.6);

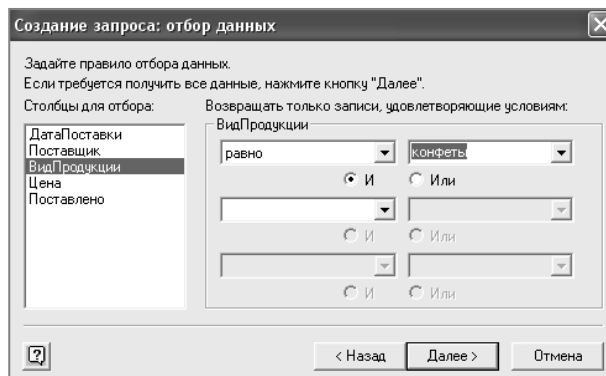


Рис. 4.6. Окно *Создание запроса: отбор данных*

- нажать кнопку *Далее*;
- указать следующий порядок сортировки: по *ВидПродукции*, затем по *ДатаПоставки*, нажать кнопку *Далее*;

- сохранить запрос, нажав кнопку *Сохранить запрос*. В открывшемся окне указать имя запроса 2 – *поставки продукции Конфеты*. Нажать кнопку *Сохранить*;
- для просмотра результата выполнения запроса в конструкторе MS Query нужно установить переключатель в строку *Просмотр или изменение данных в MS Query* в окне *Создание запроса: заключительный шаг* и нажать *Готово*.

Нужно внимательно изучить открывшееся окно (рис. 4.7).

Окно MS Query условно разделено на три области: область таблиц, область условий и область данных (или область результатов). Названия областей вполне соответствуют реализуемым ими функциям.

Необходимо обратить внимание, как интерпретирует MS Query условия, заданные пользователем на шагах создания запроса, в специальном бланке условий, очень схожем с бланком запроса в MS Access.

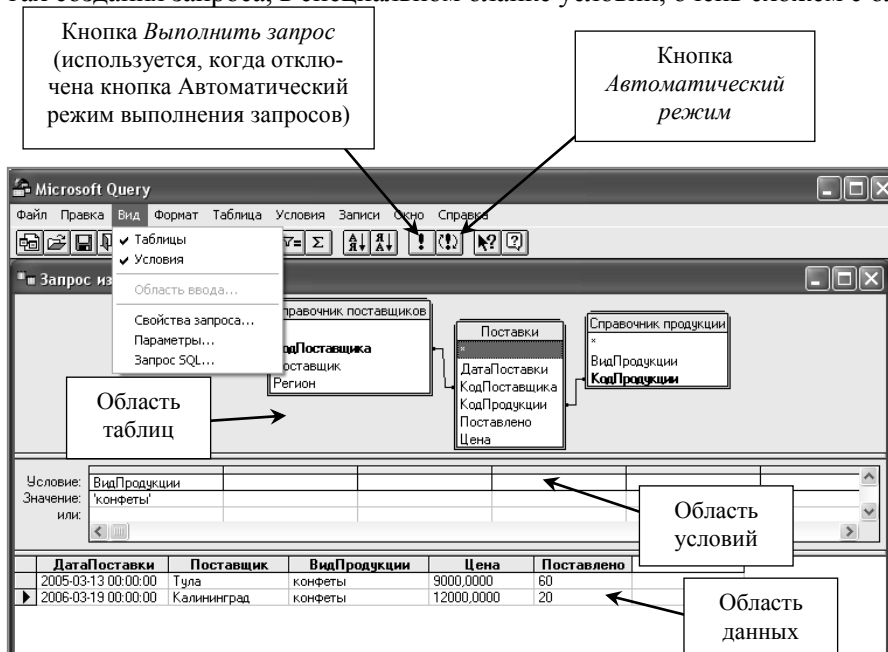


Рис. 4.7. Окно Microsoft Query

Дополнительные сведения

Для успешного выполнения ситуаций, приведенных в примерах 1 и 2, необходимо учитывать следующие сведения:

1. Если область таблиц или область условий не отображены в окне MS Query, необходимо убедиться в наличии флажков напротив слов *Таблицы* и *Условия* в команде *Вид*.
2. Чтобы предотвратить выполнение запроса до окончания ввода всех полей в таблицу данных и всех условий в область условий, кнопка *Автоматический режим* должна быть выключена. Для выполнения запроса после ввода всех данных и всех условий нажать кнопку *Выполнить запрос*.
3. Если нужно добавить новое поле в область данных, которое не было задано на предыдущих шагах создания запроса, то его можно добавить непосредственно в окне MS Query. Достаточно выполнить двойной щелчок мышью на имени поля в области таблиц.
4. Если нужно добавить еще какую-либо таблицу в область таблиц, следует выполнить команду *Таблица → Добавить таблицу...* непосредственно в окне MS Query.
5. Если какое-либо из условий было не задано на предыдущих шагах создания запроса, то его можно задать непосредственно в окне MS Query. Для этого в окне MS Query нужно выполнить команду *Условия → Добавить условие...*

Например, добавим в созданный запрос 2 (см. пример 2) еще одно условие, заключающееся в том, чтобы в список продукции с наименованием «конфеты», вносить только ту продукцию, фактическая поставка которой превышает 20 ед.

Для этого в окне MS Query нужно выполнить команду *Условия → Добавить условие...* В окне *Добавление условия* в строку *Поле* из вложенного списка выбрать имя поля, по которому создается условие (в нашем случае – *Поставлено*). В строку *Оператор* из вложенного списка выбрать нужный оператор (в нашем случае – оператор «больше») и в поле *Значение* внести значение 20 (вносить значение можно не с клавиатуры, а нажав кнопку *Значения...*, выбрать нужную цифру из списка значений и нажать *ОК*) (рис. 4.8). После заполнения полей в окне *Добавление условия* нажать кнопку *Добавить* и закрыть окно.

В результате в область условий будет внесен еще один столбец, в котором отразится созданное условие.

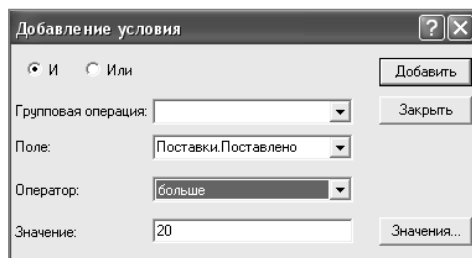


Рис. 4.8. Окно *Добавление условия* в MS Query

Чтобы сохранить запрос, доработанный в окне MS Query, нужно выполнить команду *Файл → Сохранить как*, находясь непосредственно в окне MS Query. В окне сохранения запроса (см. рис. 4.5) открыть нужную папку, ввести имя запроса: *3 – поставки продукции Конфеты, количеством больше 20* и нажать кнопку *Сохранить*.

Закреть окно MS Query.

Пример 3

Требуется самостоятельно, с помощью *Мастера запросов*, создать запрос 4 «Список фактических поставок продукции в период с 12 марта по 15 марта 2005 г. (включительно)».

Решение

Сохранить запрос с именем *4 – поставки продукции в период с 12 по 15 марта*.

Окно создания условий отбора для данного запроса представлено на рис. 4.9.

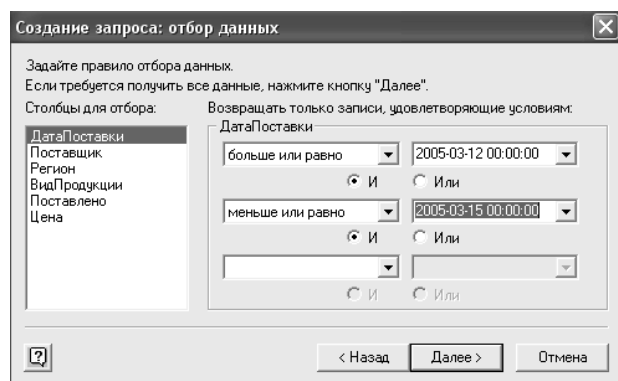


Рис. 4.9. Окно *Создание запроса: отбор данных*

Пример 4

Требуется создать сложный запрос 5 «Расчет стоимости фактической поставки каждого вида продукции, упорядоченный по стоимости».

Решение

Для выполнения данного примера нужно сделать следующее:

- выполнить команду *Данные → Импорт внешних данных → Создать запрос*;
- на вкладке *База данных* окна *Выбор источника данных* выделить имя созданного источника данных (*Поставки*), снять флажок *Использовать Мастер запросов* и нажать *ОК*;
- в открывшемся окне *Добавление таблицы* выделить таблицы, необходимые для создания запроса и добавить их в область таблиц кнопкой *Добавить*. В данный запрос следует включить все три таблицы (*Справочник поставщиков*, *Справочник продукции* и *Поставки*);
- закрыть окно *Добавление таблицы*;
- проверить, чтобы кнопка *Автоматический режим* на панели инструментов окна MS Query была отключена;
- перенести в область данных те поля, содержимое которых следует отображать по результатам запроса, выполняя двойной щелчок мышью на нужном поле в области таблиц. В данном запросе – это поля *ДатаПоставки*, *ВидПродукции*, *Поставлено*, *Цена*;
- для создания вычисляемого поля щелкнуть в *пустой заголовок столбца* в области данных и ввести выражение для расчета (выражение не должно начинаться со знака «равно» (=)): *Поставлено*Цена*;
- нажать клавишу *Enter* для завершения ввода вычисляемого поля.

Результат проделанных действий представлен на рис. 4.10.

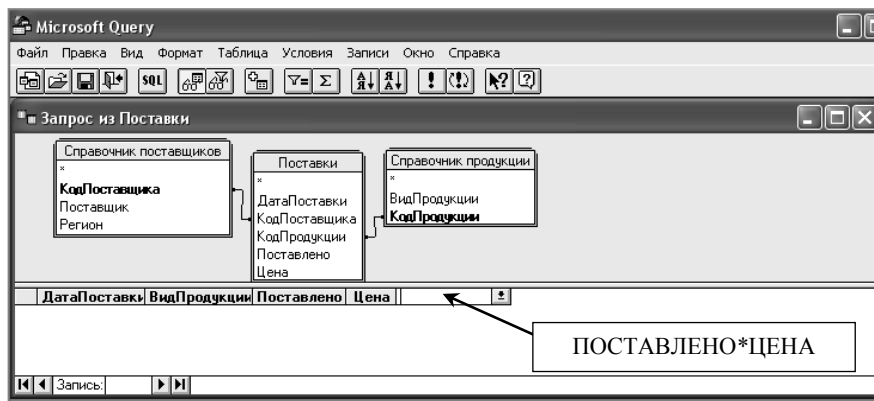


Рис. 4.10. Окно создания вычисляемого поля в MS Query

Для задания сортировки нужно выполнить следующее:

- по вычисляемому полю *Поставлено*Цена* выполнить команду *Записи → Сортировать*;
- в окне *Сортировка* в строку с именем *Столбец* из вложенного списка выбрать поле, по которому будет выполняться сортировка (поле *Поставлено*Цена*);
- задать способ сортировки: *по возрастанию*;
- нажать кнопку *Добавить* для того, чтобы добавить данную сортировку в список сортировок для данного запроса, заполненное окно сортировки представлено на рис. 4.11;

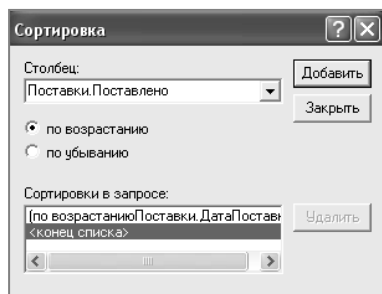


Рис. 4.11. Окно Сортировка в MS Query

- закрыть окно *Сортировка*;
- выполнить запрос, нажав кнопку *Выполнить запрос* на панели инструментов MS Query;
- сохранить запрос командой *Файл → Сохранить как* в окне MS Query. В окне сохранения запроса открыть нужную папку, ввести имя запроса *5 – расчет стоимости* и нажать *Сохранить*.

Пример 5

Нужно самостоятельно создать сложный запрос 6 «Список продукции, стоимость фактической поставки которой превышает 100 000 р., упорядоченный по стоимости фактической поставки».

Решение

В качестве подсказки можно воспользоваться данными рис. 4.12.

Сохранить запрос с именем *6 – стоимость поставки больше 100 000 р.*

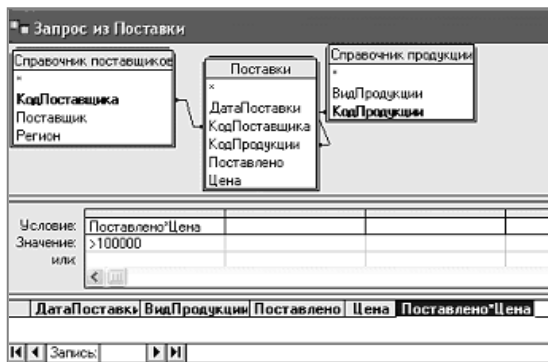


Рис. 4.12. Окно запроса с вычисляемым полем

Пример 6

Требуется создать сложный запрос 7 «Список фактических поставок продукции по поставщику, наименование которого является параметром запроса, упорядоченный по дате».

Решение

Для решения данного примера нужно сделать следующее:

- выполнить команду *Данные* → *Импорт внешних данных* → *Создать запрос*;
- на вкладке *База данных* окна *Выбор источника данных* выделить имя созданного источника данных (*Поставки*), снять флажок *Использовать Мастер запросов* и нажать *ОК*;
- в открывшемся окне *Добавление таблицы* выделить таблицы, необходимые для создания запроса и добавить их в область таблиц кнопкой *Добавить*. В данный запрос следует включить все три таблицы (*Справочник поставщиков*, *Справочник продукции* и *Поставки*);
- закрыть окно *Добавление таблицы*;
- проверить, чтобы кнопка *Автоматический режим* на панели инструментов окна MS Query была отключена;
- перенести в область данных те поля, содержимое которых следует отображать по результатам запроса, выполняя двойной щелчок мышью на нужном поле в области таблиц. В данном запросе это поля *ДатаПоставки*, *Поставщик*, *ВидПродукции*, *Поставлено*, *Цена*;
- перенести поле *Поставщик* из области таблиц в строку *Условие* области условий методом перетаскивания;
- установить курсор в строку *Значение* (напротив поля *Поставщик*) и в квадратных скобках записать параметр [Введите наименование нужного поставщика] (рис. 4.13);

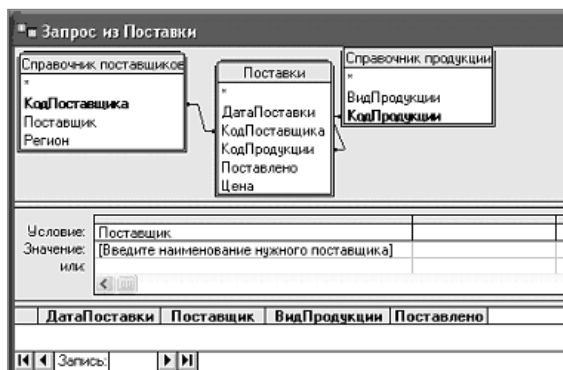


Рис. 4.13. Окно запроса с параметром


- сохранить запрос, выполнив команду *Файл* → *Сохранить как* из строки меню окна MS Query. В окне сохранения запроса открыть нужную папку, ввести имя запроса 7 – *поставки по поставщику (параметр)* и нажать кнопку *Сохранить*.

Пример 7

Необходимо самостоятельно создать сложный запрос 8 «Список продукции, поставленной в период, границы которого являются параметрами запроса».

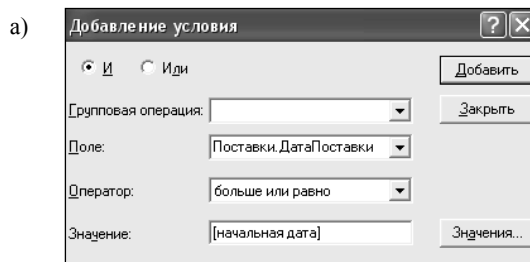
Решение

Для создания условия нужно использовать команду *Условия* → *Добавить условия* (рис. 4.14 а).

Затем нажать кнопку  *Добавить*. Установить переключатель и заполнить второе окно условия (рис. 14.4 б).

Снова нажать кнопку *Добавить*, затем – *Заккрыть*.

В качестве подсказки можно воспользоваться данными рис. 4.14 в.



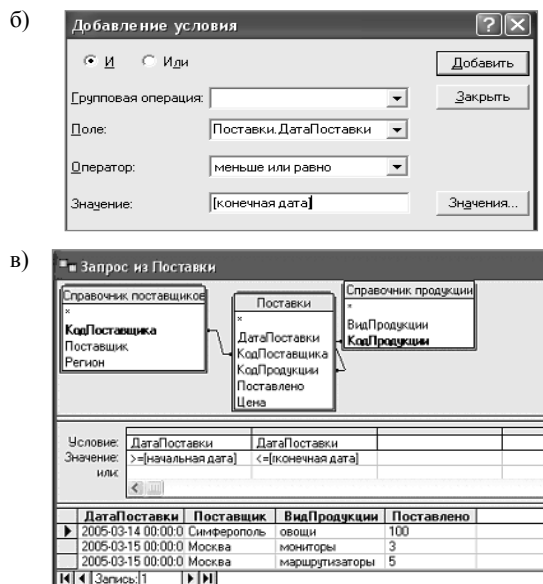


Рис. 4.14. Окно запроса с параметром

Сохранить запрос с именем δ – *поставки за период (границы периода – параметр)*.

Выполнить запрос, используя в качестве исходных данных период с 14 по 15 марта 2005 г. (включительно).

Пример 8

Требуется создать сложный запрос 9 «Общее количество поставленной продукции каждого вида». Так называемый запрос с группировкой, в которой группируются записи по полю *ВидПродукции*, а по полю *Поставлено* рассчитывается общее количество поставленной продукции данного вида.

Решение

Для создания запроса нужно сделать следующие действия:

1. Выполнить команду *Данные* → *Импорт внешних данных* → *Создать запрос*.
 2. На вкладке *База данных* окна *Выбор источника данных* выделить имя созданного источника данных (*Поставки*), снять флажок *Использовать Мастер запросов* и нажать *ОК*.
 3. В открывшемся окне *Добавление таблицы* выделить таблицы, необходимые для создания запроса, и добавить их в область таблиц кнопкой *Добавить*. В данный запрос следует включить две таблицы *Справочник продукции* и *Поставки*.
 4. Закрыть окно *Добавление таблицы*.
 5. Проверить, чтобы кнопка *Автоматический режим* на панели инструментов окна MS Query была отключена.
 6. Перенести в область данных те поля, содержимое которых следует отображать по результатам запроса, выполняя двойной щелчок мышью на нужном поле в области таблиц. В данном запросе это поля *ВидПродукции*, *Поставлено*.
 7. Выполнить двойной щелчок кнопкой мыши на поле *Поставлено* в области данных.
 8. В открывшемся окне *Изменение столбца* задать следующие параметры:
 - *Поле*, по которому подбивается итог, – поле *Поставлено* (выбрать из вложенного списка);
 - *Заголовок столбца*, в который будет помещаться рассчитанный итог – *Итого по количеству* (ввести с клавиатуры);
 - тип *групповой операции* – *Сумма* (выбрать из вложенного списка).
- Заполненное окно *Изменение столбца* изображено на рис. 4.15.

9. Нажать *ОК*.

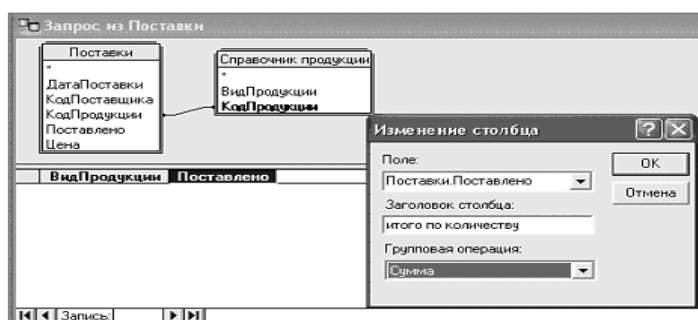


Рис. 4.15. Окно *Изменение столбца* для задания групповой операции

10. Выполнить запрос, нажав кнопку *Выполнить запрос* на панели инструментов MS Query.

11. Сохранить запрос командой *Файл* → *Сохранить как* в окне MS Query. В окне сохранения запроса открыть нужную папку, ввести имя запроса 9 – *итоговые поставки по количеству продукции* и нажать кнопку *Сохранить*.

Пример 9

Необходимо самостоятельно создать сложный запрос 10 «Общая стоимость поставок продукции за каждую дату».

Решение

В качестве подсказки можно воспользоваться данными рис. 4.16.

Выполнить запрос, нажав кнопку *Выполнить запрос* на панели инструментов MS Query.

Заккрыть окно MS Query.

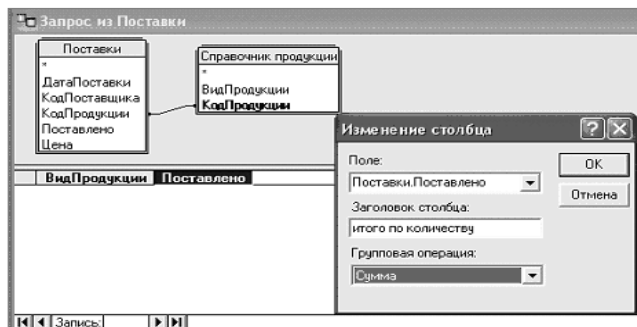


Рис. 4.16. Окно создания итогового запроса по стоимости

4.2.4. Выполнение созданных и сохраненных запросов.

Перенос результатов их выполнения на отдельные листы MS Excel

Чтобы перенести результаты созданных и сохраненных запросов на отдельные листы MS Excel, нужно выполнить следующие действия:

1. В открытой книге MS Excel перейти на новый лист и присвоить ему имя *ФактПоставки*.
2. Выполнить команду *Данные* → *Импорт внешних данных* → *Создать запрос*.
3. Снять флажок *Использовать мастер запросов* и нажать *ОК*.
4. Перейти на вкладку *Запросы* окна *Выбор источника данных*.
5. Выделить имя запроса в списке запросов (в нашем случае – *1-фактические поставки.dgy*) и нажать кнопку *Открыть*.
6. В окне MS Query выполнить команду *Файл* → *Вернуть данные в MS Excel*.
7. В открывшемся окне диалога ответить на вопрос: *Куда следует поместить данные*, установить переключатель в строку *На имеющийся лист* и указать ячейку, с которой начать вывод данных. Нажать кнопку *ОК* (рис. 4.17).

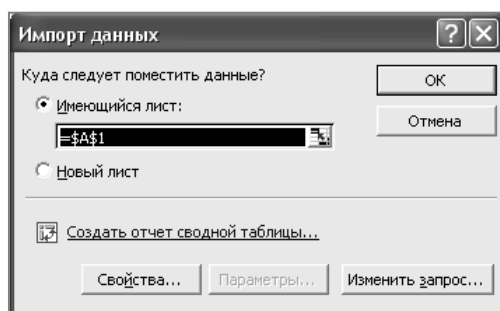


Рис. 4.17. Окно Импорт данных в MS Excel

В результате в открытую книгу Excel в указанную ячейку активного листа будет выведен результат выполнения запроса к источнику данных.

8. Аналогично переименовать другие листы Excel и перенести на них результаты выполнения оставшихся запросов. Для выполнения данного пункта воспользоваться данными табл. 4.2.

Таблица 4.2. Таблица имен листов MS Excel

Имя рабочего листа MS Excel	Имя запроса
ФактПоставки	1– фактические поставки.dqu
ПоставкиКонфет	2 – поставки продукции конфеты.dqu
Поставки(больше 20)	3 – поставки продукции конфеты и количеством больше 20.dqu
Поставки12–15марта	4 – поставки продукции в период с 12 по 15 марта.dqu
Стоимость	5 – расчет стоимости.dqu
Стоимость(больше10000)	6 – стоимость поставки больше 100 000 p. dqu
Поставщик(параметр)	7 – поставки по поставщику (параметр).dqu
Поставки(период-параметр)	8 – поставки за период (границы периода – параметр).dqu
Итог(по количеству)	9 – итоговые поставки по количеству продукции.dqu
Итог(по стоимости)	10 – общая стоимость поставок за каждую дату.dqu

Дополнительные сведения

1. Созданный и сохраненный запрос всегда можно изменить (откорректировать). Для этого нужно выполнить команду *Данные → Импорт внешних данных → Создать запрос* в окне MS Excel.

На вкладке *Запросы* окна *Выбор источника данных* выделить имя созданного вами запроса (например, *2 – поставки продукции «Конфеты»*) и нажать кнопку *Открыть*. Пройти все шаги создания запроса, изменяя и корректируя все, что посчитаете нужным. Сохранить запрос с новым именем, нажав кнопку *Сохранить запрос*. При необходимости просмотреть его в MS Query, установив переключатель в строку *Просмотр* или *Изменение данных в MS Query* и нажать кнопку *Готово*.

2. Перенести результаты выполнения запроса на лист MS Excel можно непосредственно из окна MS Query. Достаточно выполнить команду *Файл → Вернуть данные в MS Excel* в окне надстройки MS Query.

3. Чтобы очистить лист MS Excel от результатов выполнения запроса, перенесенных из MS Query, нужно установить курсор в любую ячейку диапазона результатов запроса и выполнить команду *Данные → Внешние данные → Свойства диапазона данных*. В открывшемся окне снять флажок в строке *Сохранить определение запроса* и нажать кнопку *ОК*.

В окне системного сообщения подтвердить операцию, нажав *ОК* (рис. 4.18).

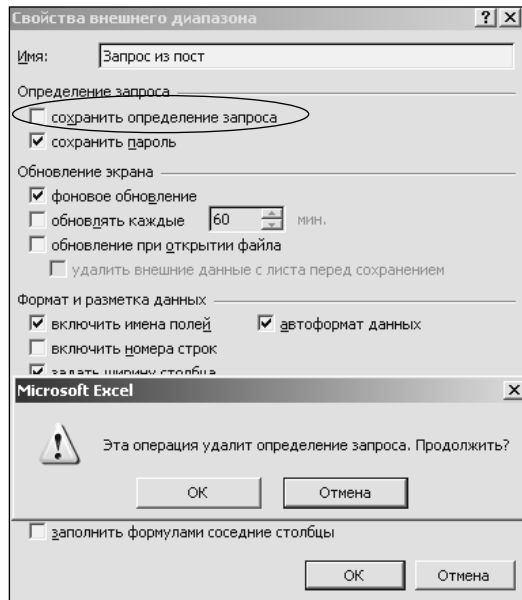


Рис. 4.18. Окно системного сообщения

Таким образом, определение запроса будет снято и результаты можно будет очистить обычным способом (выделить диапазон результатов и нажать кнопку *Delete* на клавиатуре).

4.2.5. Помещение результатов выполнения запросов в сводную таблицу MS Excel

Перенос результатов запроса в сводную таблицу в основном используется для того, чтобы создать так называемый перекрестный запрос. Причем данные для сводной таблицы подготавливаются в окне MS Query в виде запроса.

Выполним следующую задачу.

Необходимо отобразить в виде сводной таблицы общую стоимость поставок каждого вида продукции каждым поставщиком за каждую дату, предварительно создав соответствующий запрос в MS Query (в MS Query создается запрос с вычисляемым полем: стоимость поставки).

Подготовка данных для сводной таблицы в окне MS Query производится в следующем порядке:

1. Выполнить команду *Данные → Импорт внешних данных → Создать запрос*.
2. На вкладке *База данных* окна *Выбор источника данных* выделить имя созданного источника данных (*Поставки*), снять флажок *Использовать мастер запросов* и нажать *ОК*.
3. В открывшемся окне *Добавление таблицы* выделить таблицы, необходимые для создания запроса, и добавить их в область таблиц кнопкой *Добавить*. В данный запрос следует включить все три таблицы *Справочник поставщиков*, *Справочник продукции* и *Поставки*.
4. Закрыть окно *Добавление таблицы*.
5. Проверить, чтобы кнопка *Автоматический режим* на панели инструментов окна MS Query была отключена.
6. Перенести в область данных те поля, содержимое которых следует отображать по результатам запроса, выполняя двойной щелчок мышью на нужном поле в области таблиц. В данном запросе это поля *ДатаПоставки*, *ВидПродукции*, *Поставщик* и вычисляемое поле *Поставлено*Цена*.
7. Для создания вычисляемого поля щелкнуть в *пустой заголовок столбца* в области данных и ввести выражение для вычисления (выражение не должно начинаться со знака «равно» (=)): *Поставлено*Цена*. Нажать клавишу *Enter* для завершения ввода вычисляемого поля.
8. Сохранить запрос командой *Файл → Сохранить как* в окне MS Query. В окне сохранения запроса открыть нужную папку, ввести имя запроса *11 – для сводной таблицы* и нажать *Сохранить*.

Помещение результатов выполнения запросов в сводную таблицу MS Excel производится двумя способами.

1 способ

Если окно MS Query еще не закрыто, то нужно сделать следующее:

1. Выполнить команду *Файл → Вернуть данные в Microsoft Excel*.
2. В открывшемся окне диалога ответить на вопрос *Куда следует поместить данные*, установив переключатель в строку *Создать отчет сводной таблицы*. Затем нажать *ОК*.
3. В окне *Мастер сводных таблиц* нажать кнопку *Макет*.
4. Перетащить кнопки полей в нужные области сводной таблицы. В нашем задании это следующие поля:

- *ДатаПоставки* поместить в область страниц;
- *Поставщик* – в область строк;
- *ВидПродукции* – в область столбцов;
- *Поставлено*Цена* – в область данных.

Задать тип операции – *Сумма*.

Для разработки сводной таблицы воспользоваться данными рис. 4.19.

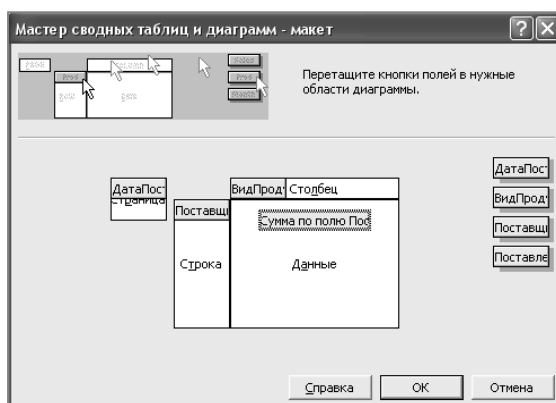


Рис. 4.19. Окно *Мастер сводных таблиц*

5. Выполнив разработку структуры сводной таблицы, нажать кнопку *Далее* для перехода к завершающему шагу создания сводной таблицы.
6. В окне *Мастер сводных таблиц – шаг 3 из 3* установить переключатель в строку *Новый лист* и нажать кнопку *Готово*.
7. Переименовать рабочий лист с полученной сводной таблицей (рис. 4.20). Назвать его *СводнаяТаблица*.
8. Сохранить все рабочие листы в книгу Microsoft Excel с именем *Импортирование1(Поставки).xls*.

Microsoft Excel - Книга1

Файл Правка Вид Вставка Формат Сервис Данные Окно Справка

Arial Cyr 10 Ж К У

G26

	A	B	C	D	E	F	G	H	I	J
1	ДатаПоставки	(Все)								
2										
3	Сумма по полю Поставлено*Цена	ВидПродукции								
4	Поставщик	консервы	конфеты	кофе	маршрутизатор	мониторы	овощи	рыба	чай	Общий итог
5	Калининград		240000	600000					3500000	840000
6	Краснодар									3500000
7	Москва				600000	1050000				1650000
8	Мурманск	150000						1050000		1200000
9	Симферополь						500000			500000
10	Тула		540000							540000
11	Хабаровск	108000								108000
12	Общий итог	258000	780000	600000	600000	1050000	500000	1050000	3500000	8338000

Рис. 4.20. Окно MS Excel со сводной таблицей

2 способ

Если окно MS Query уже закрыто, то нужно сделать следующее:

1. Выполнить команду Данные → Импорт внешних данных → Создать запрос в окне MS Excel.
2. Перейти на вкладку Запросы и выделить имя запроса, созданного для сводной таблицы (11 – для сводной таблицы.dqu).
3. Нажать кнопку Открыть.
4. Повторить действия 1–8 из первого способа.

4.2.6. Задания для самопроверки

Теперь, когда имеются необходимые знания по обработке в MS Excel внешних баз данных, требуется самостоятельно создать следующие запросы к источнику данных *Поставки*:

1. Список фактических поставок продукции за дату, значение которой является параметром запроса, упорядоченный по виду продукции.
2. Список фактических поставок продукции, поставленной после даты, значение которой является параметром запроса, упорядоченный по дате.
3. Общее количество поставок продукции, выполненных каждым поставщиком.
4. Общая стоимость поставок продукции, выполненных каждым поставщиком.
5. Общее количество продукции, поставленной за каждый год (рис. 4.21).

11- для сводной таблицы.dqu

Поставки	Справочник постав	Справочник прод
ДатаПоставки	КодПоставщика	ВидПродукции
КодПоставщика	Поставщик	КодПродукции
КодПродукции	Регион	
Поставлено		
Цена		

Year(ДатаПоставки)	Итого поставлено
2005	1528,0
2006	20,0

Рис. 4.21. Окно создания запроса 5

6. Увеличение цен на продукцию на 20%.
7. Общее количество продукции, поставленной за 2006 г. (рис. 4.22).

Поставки	Справочник постав	Справочник прод
ДатаПоставки	КодПоставщика	ВидПродукции
КодПоставщика	Поставщик	КодПродукции
КодПродукции	Регион	
Поставлено		
Цена		

Условие: Year(ДатаПоставки)

Значение: 2006

или:

Year(ДатаПоставки)	Итого поставлено
2006	20,0

Рис. 4.22. Окно создания запроса 7

8. Список продукции, поставка которой больше 30 ед. или меньше 5 ед., или продукция, поставленная поставщиком «Москва» (рис. 4.23).

Условие:	Поставлено	Поставщик	
Значение:	>30 Или <5		
или:		'Москва'	

Рис. 4.23. Вид области условий для запроса 8

9. *Список фактических поставок продукции, наименование которой начинается на букву «м», упорядоченный по дате поставки.

10. *Общая стоимость поставок продукции, поставленной за март 2005 г.

Примечание. * – запрос повышенной сложности.

4.3. Задания для самостоятельной работы

Задание 1. В MS Access требуется создать базу данных *Сервис-центр.mdb*.

Порядок выполнения данного задания содержится в методическом пособии [4].

Задание 2. Необходимо создать в MS Query новый источник данных *Сервис-центр*. В качестве источника использовать базу данных, созданную в MS Access с именем *Сервис-центр.mdb*.

Задание 3. Создать и сохранить в MS Query нижеприведенные запросы к созданному источнику данных:

- список всех услуг, предлагаемых предприятием «Сервис-центр», упорядоченный по наименованию;
- список всех клиентов центра;
- список всех заказов, упорядоченный по клиентам, а затем по дате;
- список услуг стоимостью более 10 000 р., упорядоченный по стоимости;
- список услуг, которые выполняются до 2 дней включительно или не менее, чем за неделю;
- *список клиентов, наименование которых начинается с символов «пк»;
- список услуг стоимостью более 8 000 р., которые выполняются не более, чем за 3 дня;
- список неоплаченных заказов;
- список неоплаченных заказов, а также тех, которые имеют стоимость, меньшую 5 000 р.;
- список заказов, выполненных за некоторый день, значение которого является параметром запроса;
- список заказов, выполненных за некоторый период, границы которого являются параметрами запроса;
- список оплаченных заказов, оказанных некоторому клиенту, наименование которого является параметром запроса;
- список заказов, которые еще не выполнены;
- список заказов с номерами от 10 до 20 или за текущий месяц;
- список услуг со стоимостью, увеличенной на 10% относительно исходной;
- общая стоимость услуг, оказанных каждому клиенту;
- общая стоимость оплаченных заказов по услуге каждого вида;
- общая стоимость оплаченных услуг, оказанных за каждую дату.

Задание 4. Реализовать активные запросы (с применением SQL):

- создание новой таблицы, содержащей все заказы;
- удаление заказов с датой оформления 01.06.2001;
- добавление заказов с датой оформления 01.06.2001;
- увеличение стоимости каждой услуги на 10%.

Задание 5. Создать в MS Query запросы, поместив результат их выполнения в *сводную таблицу* MS Excel (назвать листы Excel со сводными таблицами *Сводная_таблица1*, *Сводная_таблица2*):

- общая стоимость заказов, выполненных предприятием для каждого клиента по каждой услуге;
- *общее количество заказов по каждой услуге за каждый год.

Задание 6. Сохранить результаты работы в книгу *Импортирование2(Сервис-центр).xls*.

Раздел 5. ИСПОЛЬЗОВАНИЕ НАДСТРОЙКИ MS EXCEL «ПОИСК РЕШЕНИЯ» В ЭКОНОМИЧЕСКИХ РАСЧЕТАХ

5.1. Основные теоретические сведения

Средство *Поиск решения* из пакета MS Excel предназначено для выполнения сложных вычислений, которые трудно выполнить вручную. Оно позволяет находить значения в целевой ячейке, изменяя при этом

до 200 переменных, удовлетворяющих заданным критериям. По желанию пользователя результаты поиска могут быть представлены в виде отчетов разных типов, которые можно поместить в рабочую книгу.

Перед тем, как начать поиск решения, необходимо четко сформулировать решаемую проблему, т. е. выбрать входные данные и определить ограничения, чтобы надстройка MS Excel нашла осмысленное значение.

Исходные данные для запуска средства *Поиск решения* должны быть представлены в виде таблицы, которая содержит формулы, отражающие зависимости между данными таблицы.

5.2. Применение средства Поиск решения для определения оптимального количества выпускаемых изделий с максимальной прибылью

5.2.1. Постановка задачи

Рассмотрим работу средства *Поиск решения* на примере.

Для изготовления трех видов изделий А, В и С используется токарное, фрезерное, сварочное и шлифовальное оборудование. Определить, сколько изделий каждого вида следует изготовить предприятию, чтобы прибыль от их реализации была максимальной, при выполнении ряда условий:

- общий фонд времени работы каждого из типов оборудования не может быть больше, чем запланированный (заданный);
- количество выпускаемых изделий каждого вида не может быть отрицательным числом и должно быть целым.

Исходные данные к задаче представлены в виде табл. 5.1.

Таблица 5.1. Исходные данные

Производственное оборудование	Затраты времени на обработку одного изделия (станко-часы)			Общий фонд рабочего времени оборудования (ч)
	А	В	С	
Фрезерное	2	4	1	120
Токарное	1	8	6	280
Сварочное	7	4	1	240
Шлифовальное	4	6	4	360
Прибыль от продажи, усл. ед.	10	14	20	

5.2.2. Последовательность решения задачи в среде MS Excel

Решение задачи в среде MS Excel выполняется в следующем порядке:

1. Заполнение ячеек Excel соответствующими значениями (рис. 5.1)

Microsoft Excel - пример.xls

Файл Правка Вид Вставка Формат Сервис Данные Окно ?

100%

А1 B1 C1 D1 E1 F1

A B C D E F

Производственный участок Затраты времени(станко-ч) на обработку одного изделия Общий фонд рабочего времени оборудования (ч)

1

2 A B C

3 Фрезерное 2 4 1 120

4 Токарное 1 8 6 280

5 Сварочное 7 4 1 240

6 Шлифовальное 4 6 4 360

7 Прибыль от продажи, \$ 10 14 20

8

9 Решение: A B C

10 Количество изделий 0 0 0

11

12 Трудозатраты:

13 Фрезерное 0

14 Токарное 0

15 Сварочное 0

16 Шлифовальное 0

17

18 Прибыль от продажи, \$ 0

19

=СУММПРОИЗВ(\$B\$10:\$D\$10;B3:D3)

Рис. 5.1. Экран Excel для решения задачи

Ячейки B10:D10 отведены под значения неизвестных, которые требуется найти. Этим ячейкам присваиваются начальные значения (0, 0, 0). После решения задачи Excel запишет в эти ячейки найденные оптимальные значения. Вычислим значения ограничений при начальных значениях переменных. Для этого введем в ячейку B13 формулу

$$=СУММПРОИЗВ(\$B\$10:\$D\$10;B3:D3).$$

Эта формула задает величину времени работы фрезерного оборудования (станко-часы), необходимую для производства всех изделий всех видов. В ячейку E3 был заранее внесен запланированный объем времени работы фрезерного оборудования.

Ячейки B14:B16 заполняются формулами аналогично (формула копируется из ячейки B13).

Вычислим значение целевой функции (т. е. прибыли от продажи изделий) при начальных значениях неизвестных. В ячейку B18 запишем формулу вычисления общей прибыли:

$$=СУММПРОИЗВ(B10:D10;B7:D7).$$

Экран Excel в режиме представления формул показан на рис. 5.2.

	A	B	C	D	E
1	Производственный участок	Затраты времени(станко-ч) на обработку одного изделия			Общий фонд рабочего времени оборудования
2		A	B	C	
3	Фрезерное	2	4	1	120
4	Токарное	1	8	6	280
5	Сварочное	7	4	1	240
6	Шлифовальное	4	6	4	360
7	Прибыль от продажи, \$	10	14	20	
8					
9	Решение:	A	B	C	
10	Количество изделий	0	0	0	
11					
12	Трудозатраты:				
13	Фрезерное	=СУММПРОИЗВ(\$B\$10:\$D\$10;B3:D3)			
14	Токарное	=СУММПРОИЗВ(\$B\$10:\$D\$10;B4:D4)			
15	Сварочное	=СУММПРОИЗВ(\$B\$10:\$D\$10;B5:D5)			
16	Шлифовальное	=СУММПРОИЗВ(\$B\$10:\$D\$10;B6:D6)			
17					
18	Прибыль от продажи, \$	=СУММПРОИЗВ(B10:D10;B7:D7)			

Рис. 5.2. Экран Excel в режиме представления формул

2. Сохранение исходных данных в сценарий

Прежде, чем приступить к процессу поиска оптимального решения задачи, сохраним исходные данные задачи в виде сценария.

Сценарий Excel – это инструмент, позволяющий моделировать различные физические, экономические, математические и другие задачи. Он представляет собой зафиксированный в памяти компьютера набор значений ячеек рабочего листа. Используя сценарии, можно сохранить в памяти компьютера несколько наборов исходных данных так, чтобы их можно было быстро загрузить (и получить результат, соответствующий этому набору данных).

Для создания сценария выполним команду *Сервис* → *Сценарии*. В появившемся окне нажмем кнопку *Добавить* для создания сценария.

В окне *Добавление сценария* в строку *Название сценария* введем имя *Исходные данные*, а в строку *Изменяемые ячейки* запишем диапазон ячеек, содержащий исходные данные: диапазон значений затрат времени на обработку одного изделия (\$B\$3:\$D\$6) и диапазон значений общего фонда рабочего времени оборудования (\$E\$3:\$E\$6). Чтобы не писать имена диапазонов вручную, можно выделить их на листе Excel. Если данные для сценария располагаются в несмежных диапазонах, то выделим их с нажатой клавишей *Ctrl*.

Выделив диапазон ячеек, нажмем *OK*. Результат проделанных действий должен иметь вид, представленный на рис. 5.3.

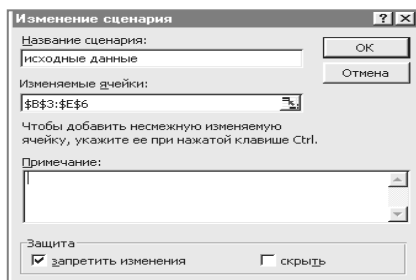


Рис. 5.3. Окно Изменение сценария

В окне *Значения ячеек сценария* можно проверить значения, которые хранятся в ячейках диапазона, который был выделен на предыдущем шаге. Проверив данные, нажмем кнопку *ОК*. Таким образом вернемся в окно *Диспетчер сценариев* и увидим имя созданного сценария. Теперь окно можно закрыть.

В дальнейшем, для того, чтобы вернуть исходные данные в задачу, достаточно будет выполнить команду *Сервис* → *Сценарии*. В списке сценариев выделить имя нужного сценария и нажать кнопку *Вывести*.

3. Заполнение окна поиска решения

Чтобы начать процесс поиска решения, выполним команду *Сервис* → *Поиск решения*. На экране появится окно *Поиск решения*.

Если такого пункта в меню *Сервис* не имеется, следует загрузить соответствующую программу-надстройку. Для этого надо выполнить команду *Сервис* → *Надстройки*. В открывшемся окне диалога установить флажок в строке *Поиск решения* (рис. 5.4).

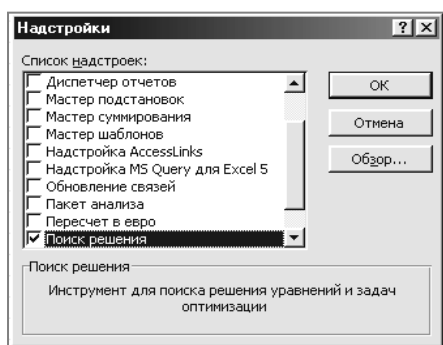


Рис. 5.4. Окно *Настройки*

Установим курсор в поле *Установить целевую ячейку* и укажем ячейку модели, значение которой должно быть изменено (максимизировано, минимизировано или приравнено к какому-либо определенному указанному значению). В нашей модели целевой будет ячейка, содержащая формулу расчета прибыли B18.

Целевая ячейка должна содержать формулу, которая прямо или косвенно ссылается на изменяемые ячейки.

С помощью переключателя *Равной* на три положения (*Максимальному значению*, *Минимальному значению* и *Значению*) зададим, соответственно, максимизацию, минимизацию или установку определенного значения целевой ячейки. В последнем случае необходимо указать число в поле *Значение*.

В данном примере необходимо установить переключатель в положение *Максимальному значению*.

В поле *Изменяя ячейки* установим ссылки на ячейки, которые будут изменяться. Сделать это можно двумя способами: ввести адреса или имена ячеек с клавиатуры или указать ячейку (диапазон ячеек) на рабочем листе с помощью мыши.

При нажатии кнопки *Предположить* автоматически выделяются ячейки, на которые есть прямая или косвенная ссылка в формуле целевой ячейки.

Введем адрес диапазона B10: D10.

Следующий этап – определение ограничений. Для этого нажмем кнопку *Добавить*. На экране появится окно диалога *Добавление ограничения* (рис. 5.5).

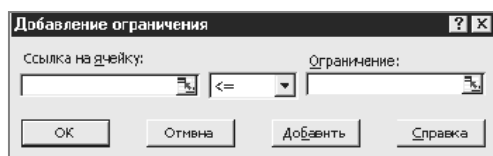


Рис. 5.5. Окно *Добавление ограничения*

В поле *Ссылка на ячейку* указывается адрес ячейки или диапазона ячеек, для которых должно действовать ограничение. В списке операторов нужно выбрать оператор из списка. В поле *Ограничение* указывается число или ссылка на какую-либо ячейку или диапазон.

Ограничения можно задать как для изменяемых ячеек, так и для целевой ячейки, а также для других ячеек, прямо или косвенно присутствующих в модели.

Если в поле *Ограничение* указана ссылка на диапазон ячеек, размер этого диапазона должен совпадать с размером диапазона, указанного в поле *Ссылка на ячейку*.

Введем первое ограничение (по условию задачи: общий фонд времени работы каждого из типов оборудо-

дования не может быть больше, чем запланированный (заданный):

$$B_{13} \leq E_3.$$

Нажмем кнопку *Добавить*, чтобы продолжить ввод ограничений.

Так как все четыре ограничения имеют один и тот же знак (\leq), то можно ввести их одной записью:

$$B_{13}:B_{16} \leq E_3:E_6.$$

Снова нажмем кнопку *Добавить* для того, чтобы ввести еще одно ограничение на целочисленность искоемых переменных (т. е. количество выпускаемых изделий каждого вида должно быть целым числом). Для задания такого ограничения выделим диапазон ячеек B10:D10, затем из списка операторов сравнения выберем слово *цел.* и далее нажмем кнопку *OK*, чтобы завершить ввод ограничений и вернуться в окно *Поиск решения*. Заданные условия появятся в списке *Ограничения*.

С помощью кнопок *Добавить* и *Изменить* можно при необходимости откорректировать заданные ограничения.

Итак, целевая ячейка, изменяемые ячейки и ограничения для нашей модели заданы (рис. 5.6).

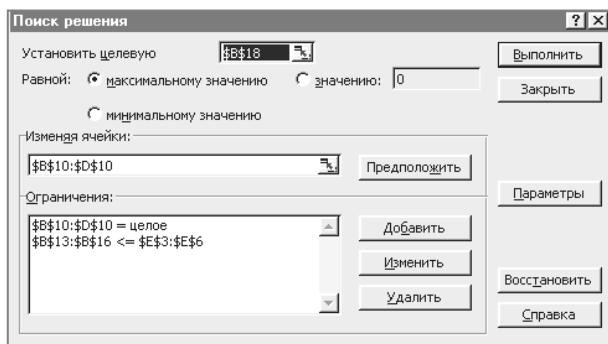


Рис. 5.6. Окно *Поиск решения*

Дальше можно *изменить* параметры поиска решения, заданные по умолчанию, а также *сохранить* созданную модель поиска решения, чтобы использовать ее в дальнейшем. Изменение параметров поиска (если это необходимо) или сохранение параметров модели выполняется в окне *Параметры поиска решения*.

Нажмем кнопку *Параметры* в окне диалога *Поиск решения*. На экране появится окно *Параметры поиска решения* (рис. 5.7).

Элементы окна:

- поле *Максимальное время* служит для ограничения времени, отпускаемого на поиск решения задачи;
- поле *Предельное число итераций* ограничивает число промежуточных вычислений;
- поля *Относительная погрешность* и *Допустимое отклонение* служат для задания точности, с которой ищется решение. Рекомендуется найти решение с величинами данных параметров, заданными по умолчанию, а затем повторить вычисления с меньшей погрешностью и допустимым отклонением;

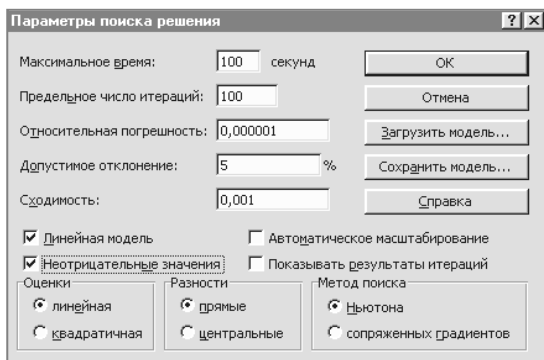


Рис. 5.7. Окно *Параметры поиска решения*

- флажок *Линейная модель* должен быть установлен в случае линейной задачи, а в случае нелинейной – сброшен;
- флажок *Неотрицательные значения* служит для включения режима нахождения только неотрицательных значений искоемых переменных;
- флажок *Показывать результаты итераций* служит для приостановки поиска решения и просмотра результатов промежуточных вычислений;
- флажок *Автоматическое масштабирование* служит для включения автоматической нормализации

входных и выходных значений, качественно различающихся по величине. Например, при максимизации прибыли в процентах по отношению к вложениям, исчисляемым в миллионах рублей.

В данном примере следует установить флажки в строках *Линейная модель* и *Неотрицательные значения*.

Ограничения поиска решения и *установленные параметры* (в данном примере, в отличие от стандартных параметров, заданных по умолчанию, нами включены параметры *Линейная модель* и *Неотрицательные значения*) можно сохранить в качестве модели. Для этого следует нажать кнопку *Сохранить модель* в окне *Параметры поиска решения*.

При нажатии этой кнопки появится окно для указания диапазона ячеек листа MS Excel, в которых будут сохранены параметры модели (рис. 5.8).

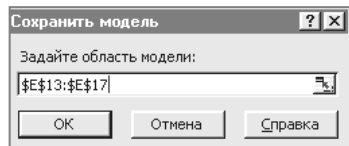


Рис. 5.8. Указание диапазона ячеек для сохранения параметров модели

На рабочем листе выделим диапазон ячеек, задающих область модели. Число выделенных ячеек должно равняться числу ограничений модели плюс три. Если выделено недостаточное количество ячеек, то Excel посоветует выбрать другой диапазон. Если выделена одна ячейка, диапазон будет выбран автоматически. В нашем случае щелкнем ячейку E13 и нажмем *ОК*.

Вернувшись в окно *Параметры поиска решения*, нажмем кнопку *ОК*, чтобы подтвердить выбранные параметры и вернуться в окно *Поиск решения*.

После того, как все параметры и ограничения заданы, осталось только инициировать поиск. Нажмем кнопку *Выполнить* в окне диалога *Поиск решения*. По мере поиска, отдельные его шаги будут отображаться в строке состояния. Когда поиск закончится, в таблицу будут внесены новые значения и на экране появится окно, сообщающее о завершении операции (рис. 5.9).

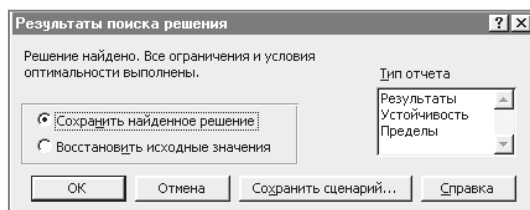


Рис. 5.9. Окно результатов поиска

4. Анализ результатов поиска и сохранение их в сценарий

Всегда нужно внимательно читать информацию, выдаваемую в данном окне. Обязательно следует убедиться, что поиск нашел подходящее решение. Об этом будет свидетельствовать соответствующая запись в окне *Результаты поиска решения*.

Поскольку полученные значения удовлетворительны, установим безымянный переключатель в положение *Сохранить найденное решение*, тогда таблица будет обновлена. Отменить результаты поиска можно, установив переключатель в положение *Восстановить исходные значения*.

В случае, если поиск закончился удачно, можно указать, какие отчеты вставить в рабочую книгу. Для этого в списке *Тип отчета* выделяем название нужного типа отчета (или несколько названий), удерживая нажатой клавишу *Ctrl*. Они будут вставлены на отдельных листах в рабочую книгу перед листом с исходными данными. Предлагаемые отчеты содержат следующую информацию:

- отчет *Результаты* содержит сведения о начальных и текущих значениях целевой ячейки и изменяемых ячеек, а также о соответствии значений заданным ограничениям;
- отчет *Устойчивость* отражает найденный результат, а также нижние и верхние предельные значения для изменяемых ячеек;
- отчет *Пределы* показывает зависимость решений от изменения формулы или ограничений.

Когда решение найти невозможно, Excel выводит соответствующее сообщение в окне диалога *Результаты поиска решения*. В этом случае возможность создать отчет отсутствует, список *Тип отчета* становится недоступным.

Сохраним сценарий найденного решения. Для этого нажмем кнопку *Сохранить сценарий* в окне *Результаты поиска решения*. В открывшемся окне *Сохранение сценариев* введем имя сценария – *Решение1* и нажмем кнопку *ОК*. Затем – еще раз *ОК*.

Выполнив все вышеуказанные действия, можно увидеть на листе Excel решение задачи, которое будет находиться в ячейках B10:D10 (рис. 5.10).

Производственный участок	Затраты времени(станко-ч) на обработку одного изделия			Общий фонд рабочего времени оборудования (ч)
	A	B	C	
1				
2				
3	Фрезерное	2	4	120
4	Токарное	1	8	280
5	Сварочное	7	4	240
6	Шлифовальное	4	6	360
7	Прибыль от продажи, \$	10	14	20
8				
9	Решение:	A	B	C
10	Количество изделий	28	0	42
11				
12	Трудозатраты:			
13	Фрезерное	98		1120
14	Токарное	280		3
15	Сварочное	238		ИСТИНА
16	Шлифовальное	280		ИСТИНА
17				100
18	Прибыль от продажи, \$			
19				

Рис. 5.10. Результаты решения

Разберем подробно, какие значения содержат ячейки, в которых сохранены параметры модели (табл. 5.2).

Таблица 5.2. Параметры модели и их назначение

Ячейка	Отображаемое значение	Формула	Назначение формулы
E13	1120	=МАКС(\$B\$18)	Указана целевая ячейка и что она должна достигать максимального значения
E14	3	=СЧЕТ(\$B\$10:\$D\$10)	Указан диапазон изменяемых ячеек
E15	ИСТИНА	=B\$13:B\$16<=Лист1!\$E\$3:\$E\$6	Указано первое ограничение на количество ресурсов
E16	ИСТИНА	=B\$10:\$D\$10=ЦЕЛОЕ(\$B\$10:\$D\$10)	Указано второе ограничение – целочисленность искомых переменных
E17	100	= { 100:100:0,000001:0,05:ИСТИНА:ЛОЖЬ:ЛОЖЬ:1:1:1:0,001:ИСТИНА }	Зафиксированы параметры поиска, устанавливаемые в окне <i>Параметры поиска решения</i>

Сохраним результаты решения задачи на листе с именем *ПроизводствоИзделий* рабочей книги *Поиск решения.xls*.

5.2.3. Контрольные вопросы

1. В чем состоит назначение средства *Поиск решения*? Как производится вызов окна *Поиск решения*?
2. Какие элементы имеет окно *Поиск решения*?
3. Что такое целевая ячейка и как задается задача ее оптимизации?
4. Какие ячейки в Excel называются изменяемыми? В чем их связь с другими ячейками? Какие есть способы задания изменяемых ячеек для средства *Поиск решения*?
5. Какие виды ограничений можно задать для средства *Поиск решения*? Поясните технологию ввода ограничений.
6. Каковы основные параметры средства *Поиск решения* и в чем их назначение?
7. Каково назначение параметра *линейная модель*?
8. Какой параметр необходимо устанавливать для задания неотрицательности искомых переменных?
9. Как выполняется сохранение и загрузка параметров модели?
10. Как выполняется анализ решения, полученного с помощью надстройки *Поиск решения*?
11. Каким образом создаются и сохраняются сценарии?
12. Каково принципиальное отличие «сохранения параметров модели» от «сохранения сценариев»?

5.3. Задачи для самостоятельного решения

Задача 1. Структура производства молочной продукции

Постановка задачи

Продукцией городского молочного завода является молоко, кефир и сметана, расфасованные в пакеты. Требуется определить, какую продукцию и в каком количестве следует ежедневно изготавливать заводу, чтобы прибыль от ее реализации была максимальной. Причем молоко и кефир производятся и разливаются на оборудовании одного и того же типа, а сметана – на оборудовании другого типа. Исходные данные к задаче представлены в табл. 5.3.

Таблица 5.3. Исходные данные к задаче 1

Тип продукции	Затраты времени по расфасовке 1 т продукции, ч	Общий фонд рабочего времени оборудования, ч	Прибыль от реализации 1 т продукции, усл. ед.	Затраты молока на производство 1 т продукции, т
Молоко	0,16	240	96	1,18
Кефир	0,18	240	110	1,2
Сметана	2,96	88	499,5839	5,45

Необходимо выполнить следующее:

1. Определить, какую продукцию и в каком количестве следует ежедневно производить заводу, чтобы прибыль от ее реализации была максимальной. При этом должен выполняться ряд условий:

- количество производимой продукции не может быть отрицательным числом;
- всего завод может использовать молока не более 1020 т;
- всего завод должен производить молока не менее 190 т;
- общий фонд времени работы оборудования по расфасовке молока и кефира составляет 240 ч;
- общий фонд времени работы оборудования по расфасовке сметаны составляет 88 ч.

2. Сохранить исходные данные (ячейки, выделенные серым цветом, на рис. 5.11) в виде сценария с именем *Исходные данные*.

3. Сохранить результаты вычислений в сценарии с именем *Решение*.

4. Оформить результаты решения задачи с применением цветового оформления диапазонов и ячеек. Сохранить результаты вычислений на листе *МолочныйЗавод* в книге Excel с именем *Поиск решения.xls*.

Схема решения

Оформление исходных данных для поиска показано на рис. 5.11.

Результаты решения представлены на рис. 5.12.

Результаты решения

Оптимальный план производства молока – 190 т, кефира – 528 т, сметаны – 30 т. Максимальная прибыль – 91188 усл. ед.

Файл Правка Вид Вставка Формат Сервис Данные Окно 2												
<div>📁 📄 🖨️ 🔍 🔄 ⏏️ ✂️ 📋 📌 📧 ↶ ↷ 🧮 📊 ⬆ ⬆ ⬆ 📈 📉 📶 100% 🛡️</div> <div>Аrial Cyr 10 Ж К U ☰ ☷ ☹ ☺ % , ‰ ‰ ☰ ☷ ☹ ☺</div>												
N14 =												
	A	B	C	D	E	F	G	H	I	J	K	
1	Тип продукции		Затраты времени по расфасовки 1 т продукции	Общий фонд рабочего времени оборудования (ч)	240	Прибыль от реализации 1 т продукции, у.е.	Затраты молока на производство 1 т продукции					
4	Молоко		0,16			96		1,18				
5	Кефир		0,18			110		1,2				
6	Сметана		2,96		88	499,5839		5,45				
7	Всего завод может использовать молока, не более, т						1020					
8	Всего завод должен производить молока, не менее, т						190					
9												
10	Решение:											
11	Завод будет ежедневно производить				При этом требуется молока, т:				0			
12	молока	кефира	сметаны	Рабочее время - молоко и кефир, ч:				0				
13	0	0	0	Рабочее время - сметана, ч:				0				
14												
15	Целевая функция:							0				
16												

Рис. 5.11. Оформление задачи на листе Excel

Microsoft Excel											
Файл Правка Вид Вставка Формат Сервис Данные Окно ?											
<div>Иконки инструментов</div>											
Arial Cyr 10 Ж К Ч [Форматирование]											
O22 =											
	A	B	C	D	E	F	G	H	I	J	K
1	Тип продукции		Затраты времени по расфасовки 1 т продукции	Общий фонд рабочего времени оборудования (ч)	Прибыль от реализации 1 т продукции, у.е.	Затраты молока на производство 1 т продукции					
2											
3											
4	Молоко		0,16	240	96	1,18					
5	Кефир		0,18		110	1,2					
6	Сметана		2,96	88	499,5839	5,45					
7	Всего завод может использовать молока, не более, т				1020						
8	Всего завод должен производить молока, не менее, т				190						
9											
10	Решение:										
11	Завод будет ежедневно производить				При этом требуется молока, т.				1020		
12	молока	кефира	сметаны	Рабочее время - молоко и кефир, ч.				125			
13	190	528	30	Рабочее время - сметана, ч.				88			
14											
15	Целевая функция:								91188		
16											
17											
18											
Лист1 Лист2 Лист3 /											

Рис. 5.12. Результаты решения

Задача 2. Структура производства с уменьшением нормы прибыли

Постановка задачи

Предприятие выпускает телевизоры, стерео- и акустические системы, используя общий склад комплектующих. Исходные данные приведены в табл. 5.4.

Таблица 5.4. Исходные данные к задаче 2

Наименование комплектующего	Запас комплектующих на складе, шт.	Норма расхода комплектующих на одно изделие, шт.		
		телевизор	стерео	акустическая система
Шасси	450	1	1	0
Кинескоп	250	1	0	0
Динамик	800	2	2	1
Блок питания	450	1	1	0
Электрическая плата	600	2	1	1
Прибыль с единицы изделия, тыс. р.		75	50	35
Примерные объемы выпуска изделий, ед.		100	100	100

Необходимо выполнить следующее:

1. Найти оптимальные размеры объемов выпуска изделий каждого вида (объемы выпуска *не* могут быть отрицательными), при которых общая прибыль от реализации всех выпускаемых изделий (с учетом влияния коэффициента отдачи $K_{отд} = 0,9$) будет наибольшей с учетом выполнения ограничений по запасу комплектующих на складе. Значение коэффициента отдачи заранее внести в ячейку на листе Excel с заголовком *Коэффициент отдачи*.

Примечание. Коэффициент отдачи показывает изменение прибыли с ростом объема производства изделий в связи с дополнительными затратами на сбыт. Чтобы учесть влияние коэффициента отдачи при записи формулы для расчета общей прибыли, получаемой предприятием от реализации всей продукции, нужно данный коэффициент записать в виде показателя степени для чисел, отражающих объемы выпуска изделий определенного вида.

Пример

Если предприятие выпускает 50 магнитофонов с удельной прибылью 10 000 р., то прибыль от реализации всех магнитофонов с учетом коэффициента отдачи $K_{отд} = 0,8$ составит:

$$10\,000 * 50^{0,8} = 228652,5 \text{ р.}$$

2. Подготовить и сохранить ряд вариантов исходных данных в соответствующих сценариях (табл. 5.5).

Таблица 5.5. Имена сценариев и их содержание

Имя сценария	Содержание сценария
Исход_данные	Исходные значения примерных объемов выпуска изделий
Склад1	Исходный вариант состояния склада (запаса комплектующих на складе)
Комплектация1	Исходный вариант комплектации изделий различных видов комплектующими различных типов
Склад2	Тут всех комплектующих на 30% больше, чем в сценарии «Склад 1»
Комплектация2	В отличие от сценария «Комплектация 1», тут к телевизору требуется один динамик (вместо двух), однако в стереосистеме требуются две платы (вместо одной)

3. Решить задачу для различных вариантов состояния склада, комплектующих (загружая соответствующий сценарий) и коэффициента отдачи, *сохраняя* для каждого из вариантов *параметры модели и сценарий решения*.

Варианты состояния склада и комплектующих приведены в табл. 5.6.

Таблица 5.6. Варианты сценариев результатов решения задачи

Имя сценария результатов решения	Имена сценариев состояния склада и комплектующих	Коэффициент отдачи
Реш1	Склад1, Комплектация1	0,9
Реш2	Склад1, Комплектация1	1 (Линейная модель!!!)
Реш3	Склад1, Комплектация2	0,9
Реш4	Склад2, Комплектация1	0,9
Реш5	Склад2, Комплектация2	0,9

4. Оформить результаты решения задачи с применением цветового оформления диапазонов и ячеек. Сохранить результаты вычислений на листе с именем *БытоваяЭлектроника* в книге Excel с именем *Поиск решения*.

Схема решения

Оформление решения представлено на рис. 5.13.

А	В	С	Д	Е	Г	Н
1	2	3	4	5	6	7
Наименование комплектующего	Запас комплектующих на складе	Телевизор	Стерео	Акк. Сист	Расход комплектующих	
Шасси	450	1	1	0	0	
Кинескоп	250	1	0	0	0	
Динамик	800	2	2	1	0	
Блок пит.	450	1	1	0	0	
Электр. плата	600	2	1	1	0	
Прибыль с единицы изделия (тыс. руб)		75	50	35	1	
Примерные объемы выпуска изделий		0	0	0		
		Коэффициент отдачи				
Общая прибыль				1		

Рис. 5.13. Оформление задачи на листе MS Excel

Результаты решения

Ответы для различных вариантов исходных данных представлены в табл. 5.7.

Таблица 5.7. Ответы задачи для различных вариантов исходных данных

Имя сценария	Телевизоры	Стерео	Акустическая система	Общая прибыль
Реш1	160	200	80	14 917 р.
Реш2	200	200	0	25 000 р.
Реш3	236	5	118	13 025 р.
Реш4	232	220	116	19 031 р.
Реш5	316	5	158	16 874 р.

Задача 3. График занятости персонала

Постановка задачи

Парк отдыха обслуживается семью группами сотрудников (группы обозначены А, Б, ..., Ж). Признак распределения на группы – разные выходные дни. Выходных дней для каждой группы – не мене двух, выходные дни следуют подряд. Один сотрудник входит только в одну группу.

Известна потребность в сотрудниках в каждый из дней (потребность различна и известна, например, на основе статистических данных о средней посещаемости парка в каждый из дней недели).

Все сотрудники имеют одинаковый размер недельной оплаты, который не зависит от графика работы и равен 40 тыс. р.

Исходные данные к задаче представлены в табл. 5.8.

Таблица 5.8. Исходные данные к задаче 3

График	Выходные дни	Примерное количество сотрудников в группе, чел.	Вс	Пн	Вт	Ср	Чт	Пт	Сб
А	Вс, Пн	4	0	0	1	1	1	1	1
Б	Пн, Вт	4	1	0	0	1	1	1	1
В	Вт, Ср	4	1	1	0	0	1	1	1
Г	Ср, Чт	6	1	1	1	0	0	1	1
Д	Чт, Пт	6	1	1	1	1	0	0	1
Е	Пт, Сб	4	1	1	1	1	1	0	0
Ж	Сб, Вс	4	0	1	1	1	1	1	0
	Общее число сотрудников	Сумма по столбцу							
Потребность в сотрудниках в каждый из дней, чел.			22	17	13	14	15	18	24

Необходимо выполнить следующее:

1. Подобрать такую численность сотрудников в каждой группе, чтобы добиться минимизации затрат на оплату труда всех сотрудников за неделю при выполнении ряда условий:

- число в сотрудников, работающих в каждый из дней недели, не может быть *меньше потребности* в них в этот день;
- число сотрудников в группе не может быть *отрицательным*, однако может быть *равным нулю*;
- число сотрудников должно быть *целым*.

Важной особенностью этой задачи является наличие нескольких оптимальных решений, каждое из которых обеспечивает достижение целевой функции при выполнении всех ограничений.

2. Методом последовательного расчета (т. е. повторного применения *Поиска решения* к уже найденному ответу) найти оптимальные решения и сохранить их путем копирования в таблицу следующего вида (табл. 5.9).

Таблица 5.9. Таблица записи результатов решения задачи

График	Численность сотрудников в группе, чел.									
	при первом решении	при втором решении	при третьем решении	при четвертом решении	при пятом решении	при шестом решении	при седьмом решении	при восьмом решении	при девятом решении	при десятом решении
А										
Б										
В										
Г										
Д										
Е										
Ж										

3. Данные, записанные в таблицу, позволят выбрать ответ по группировке сотрудников, наиболее подходящей с точки зрения дополнительных условий (которые не заданы, но могут быть заданы) в качестве дополнительных ограничений, а именно:

- подобрать решение задачи, которое обеспечивает как можно большее количество сотрудников, работающих по графику Д;
- подобрать решение задачи, которое обеспечивает как можно больший резерв сотрудников в каждый из дней;

• подобрать решение задачи, которое обеспечивает сокращение групп (т. е. ликвидацию как можно большего числа графиков работы сотрудников).

4. Подобрать решение задачи, которое обеспечивает большее, чем требуется общее число сотрудников в наиболее напряженные дни работы парка (резерв сотрудников, работающих в эти дни).

5. Оформить результаты решения задачи с применением цветового оформления диапазонов и ячеек. Сохранить результаты вычислений на листе с именем *ГрафикЗанятости* в книге Excel с именем *Поиск решения*.

Схема решения

Оформление решения представлено на рис. 5.14.

График	Выходные дни	Примерное количество сотрудников в группе	Вс	Пн	Вт	Ср	Чт	Пт	Сб
А	Воскрес., понедельник	4	0	0	1	1	1	1	1
Б	Понедельник, вторник	4	1	0	0	1	1	1	1
В	Вторник, среда	4	1	1	0	0	1	1	1
Г	Среда, четверг	6	1	1	1	0	0	1	1
Д	Четверг, пятница	6	1	1	1	1	0	0	1
Е	Пятница, суббота	4	1	1	1	1	1	0	0
Ж	Суббота, воскресенье	4	0	1	1	1	1	1	0
Всего: СУММ(C2:С8)									
Потребность в сотрудниках в каждый из дней			22	17	13	14	15	18	24
Фактическое наличие сотрудников в каждый из дней									
Недельная оплата работнику:			40р.						
Общая недельная зарплата всех сотрудников:			0р.						

Рис. 5.14. Оформление задачи на листе MS Excel

Примечание. Прежде, чем применять *Поиск решения*, нужно сохранить исходные данные (ячейки, выделенные серым цветом (см. табл. 5.8)) в виде сценария с именем *Исходные данные*.

Результаты решения

Результаты решения задачи представлены в таблицах 5.10–5.12.

Таблица 5.10. Ответы для пункта 2 задачи 3

График	Численность сотрудников в группе, чел.									
	при первом решении	при втором решении	при третьем решении	при четвертом решении	при пятом решении	при шестом решении	при седьмом решении	при восьмом решении	при девятом решении	при десятом решении
А	3	3	3	2	2	3	2	2	3	2
Б	4	4	5	6	6	5	6	6	5	4
В	8	8	7	6	6	7	6	6	6	8
Г	3	3	3	5	5	4	3	4	5	3
Д	7	6	7	5	5	5	7	6	5	7
Е	0	1	0	0	1	1	0	1	1	0
Ж	0	0	0	1	0	0	1	0	0	1

Общее число сотрудников в каждом из вариантов равно 25 и суммарные затраты на оплату труда всех сотрудников в неделю составляют 1 млн р.

Таблица 5.11. Ответ для пункта 4 задачи 3 (по численности сотрудников в группе)

График	Численность сотрудников в группе, чел.
А	1
Б	5
В	8
Г	4
Д	7
Е	1
Ж	0

Таблица 5.12. Ответ для пункта 4 задачи 3 (по общему числу сотрудников в каждый из дней недели)

Вс	Пн	Вт	Ср	Чт	Пт	Сб
25	17	13	17	15	18	25

Общее число сотрудников равно 26. Суммарные затраты на оплату труда всех сотрудников в неделю составляют 1 млн 40 тыс. р.

Задача 4. Перевозки с долевым участием нескольких перевозчиков

Постановка задачи

Руководство фирмы приняло решение о том, что в течение определенного периода (например, года) будет осуществлять перевозки групп своих сотрудников в некоторое количество пунктов (в данном случае – пять) силами нескольких (в данном случае трех) фирм-перевозчиков. Причем общее число перевозок должно быть распределено между этими перевозчиками в соотношении 2:3:2.

В качестве исходных данных известно следующее:

- доля перевозок каждого перевозчика в общем числе перевозок;
- стоимость одной перевозки каждым перевозчиком в каждый из пяти пунктов;
- требуемое количество перевозок (на планируемый период) в каждый из пяти пунктов.

Исходные данные к задаче представлены в таблицах 5.13–5.15.

Таблица 5.13. Стоимость перевозок в пункты, усл. ед.

Перевозчик	Пункт А	Пункт В	Пункт С	Пункт D	Пункт Е
1	24	16	8	10	14
2	21	15	7	12	16
3	23	14	7	14	12

Таблица 5.14. Требуемое количество перевозок в различные пункты

Пункт	А	В	С	D	Е	Требуемое итоговое количество перевозок
Число перевозок	10	15	20	10	15	Сумма перевозок по всем пунктам

Таблица 5.15. Расчетное количество перевозок перевозчиками в соответствии с их долей

Перевозчик	Доля	Число перевозок
1	2	
2	3	
3	2	
Итого	Сумма всех долей	Сумма по столбцу «Число перевозок»

Число перевозок по каждому перевозчику рассчитывается по формуле

$$\text{Число перевозок} = \text{ОКРУГЛ}(\text{сумма перевозок по всем пунктам (табл. 5.14)} / \text{сумма всех долей (табл. 5.15)} * \text{доля (табл. 5.15)} ; 0).$$

Необходимо выполнить следующее:

1. Соответственно перед начальником транспортного отдела, планирующим перевозки, стоит задача минимизации затрат на перевозки при выполнении ряда условий:

- количество перевозок, выполненных каждым перевозчиком, должно быть равно требуемому числу перевозок, рассчитанному в соответствии с его долей;
- количество перевозок, выполненных в каждый пункт, должно быть точно равно требуемому количеству перевозок в данный пункт;
- число перевозок каждым перевозчиком в каждый пункт должно быть целым числом;
- число перевозок каждым перевозчиком в каждый пункт должно быть неотрицательным.

2. Выполнив решение задачи, сохранить ответ в виде сценария с именем *Решение1*.

3. В любую свободную ячейку листа Excel (например, ячейка Н5 на рис. 5.15) записать формулу проверки равенства значений требуемого числа перевозок и фактического числа перевозок, полученного в ходе решения, в следующем виде:

ЕСЛИ (требуемое число перевозок = фактическое число перевозок;" "; "Требуемое количество перевозок НЕ РАВНО расчетному (найденному по долям перевозчиков)").

4. Задать *красный* цвет шрифта в этой ячейке.

5. Изменить исходные данные по задаче, т. е. изменить соотношение долей перевозчиков (табл. 5.16).

Таблица 5.16. Измененные исходные данные к задаче 4

Перевозчик	Доля
1	7
2	2
3	3
Итого	Сумма всех долей

6. Применить надстройку *Поиск решения* и убедиться, что поиск не может найти подходящего решения, так как требуемое число перевозок (рассчитанное по долям) не совпадает с количеством, полученным в ходе решения. Это связано с округлением, которое было применено при расчете количества перевозок в соответствии с долями.

7. Попытаться ответить на вопрос: *Как в этой ситуации поступить начальнику транспортного отдела?*

8. Подкорректировать доли перевозчиков (задать долю одного из перевозчиков дробным числом, например, первому перевозчику задать долю не 7, а 7,1). Решить задачу, применив поиск решения. Сохранить сценарий по второму решению с именем *Решение2*.

9. Оформить результаты решения задачи с применением цветового оформления диапазонов и ячеек. Сохранить результаты вычислений на листе с именем *Перевозки* в книге Excel с именем *Поиск решения*.

Схема решения

Оформить лист Excel в соответствии с рис. 5.15.

The screenshot shows an Excel spreadsheet with the following structure:

- Table 1 (Costs):** Rows 4-7, Columns A-E. Headers: Перевозчик, А, В, С, D, E. Data: (1,24,16,10,14), (2,21,15,7,12), (3,23,14,7,14).
- Table 2 (Required Quantities):** Rows 19-20, Columns A-F. Headers: Пункт, А, В, С, D, E, Итого. Data: (10,15,20,10,15,70).
- Table 3 (Solution):** Rows 23-27, Columns A-F. Headers: Перевозчик, А, В, С, D, E, Итого по перевозчику. Data: (1,0,0,10,10,0,20), (2,10,10,10,0,0,30), (3,0,5,0,0,15,20), (Итого,10,15,20,10,15,70).
- Table 4 (Total Cost):** Row 29, Column A. Formula: =СУММ(G24:G26).
- Formulas:**
 - Row 10: =ОКРУГЛ(С3:G20)/С3:G20
 - Row 11: =СУММ(B11:B13)
 - Row 14: =СУММ(C11:C13)
 - Row 18: =СУММ(B20:F20)
 - Row 22: =ЕСЛИ(G20=C14;"Требуемое количество перевозок НЕ РАВНО расчетному (найденному по долям перевозчиков)";)

Рис. 5.15. Оформление задачи на листе Excel

Результаты решения

В таблицах 5.17 и 5.18 приведены результаты *Решения1* и *Решения2*.

Таблица. 5.17. Данные *Решения1*

Перевозчик	А	В	С	Д	Е	Итого
1	0	0	10	10	0	20
2	10	10	10	0	0	30
3	0	5	0	0	15	20
Итого	10	15	20	10	15	70
Полученная общая стоимость перевозок			860			

Таблица 5.18. Данные Решения2

Перевозчик	A	B	C	D	E	Итого
1	0	0	18	10	13	41
2	10	0	2	0	0	12
3	0	15	0	0	2	17
Итого	10	15	20	10	15	70
Полученная общая стоимость перевозок			884			

Задача 5. Раскрой металлического листа по длине*Постановка задачи*

Из листового проката нужно выкроить заготовки четырех видов. Один лист длиной 184 см можно разрезать на заготовки длиной 45 см, 50 см, 65 см и 85 см. Способы разреза одного листа на заготовки и величина отходов при каждом способе приведены в табл. 5.19. Данные для табл. 5.19 рассчитаны вручную, т. е. вручную просчитаны те или иные варианты раскрой. Определить, какое количество листов по каждому из способов следует разрезать, чтобы получить нужное количество заготовок данного вида при минимальных общих отходах.

Примечание. Количество листов, используемых для изготовления заготовок любого из четырех видов должно быть *целым и неотрицательным*.

Исходные данные к задаче представлены в таблицах 5.19–5.20.

Таблица 5.19. Исходные данные к задаче 5

Длина заготовки, см	Количество заготовок, выкраиваемых из одного листа при разрезе способом, шт.												Требуемое количество заготовок, шт.	Длина листа, см
	№ 1	№ 2	№ 3	№ 4	№ 5	№ 6	№ 7	№ 8	№ 9	№ 10	№ 11	№ 12		
45	4	2	2	2	1	1	1	1	0	0	0	0	900	184
50	0	1	0	0	2	0	1	1	3	2	1	0	400	
65	0	0	1	0	0	2	1	0	0	1	2	1	200	
85	0	0	0	1	0	0	0	1	0	0	0	1	900	

Таблица 5.20. Полезное использование части листа длиной, см

Длина заготовки, см	Количество заготовок, выкраиваемых из одного листа при разрезе способом, шт.											
	№ 1	№ 2	№ 3	№ 4	№ 5	№ 6	№ 7	№ 8	№ 9	№ 10	№ 11	№ 12
45	180	90	90	90	45	45	45	45	0	0	0	0
50	0	50	0	0	100	0	50	50	150	100	50	0
65	0	0	65	0	0	130	65	0	0	65	130	65
85	0	0	0	85	0	0	0	85	0	0	0	85
Итого	180	140	155	175	145	175	160	180	150	165	180	150
Отходы с одного листа, см	4	44	29	9	39	9	24	4	34	19	4	34

Необходимо выполнить следующее:

1. Найти решение задачи при *минимизации общих отходов*.

Решение задачи схематично представлено в виде таблиц 5.21–5.26.

Таблица 5.21. Количество листов, раскроенных каждым из способов, шт.

Способ раскроя	1	2	3	4	5	6	7	8	9	10	11	12	Всего
Количество листов	0	0	0	0	0	0	0	0	0	0	0	0	

Таблица 5.22. Количество полученных заготовок, шт.

Длина заготовки, см	Итого
45	
50	
65	
85	

Общие отходы (от числа листов, раскроенных каждым из способов), см.
Всего:

2. Скопировать результаты решения задачи в табл. 5.23–5.24.

Таблица 5.23. 1 вариант решения (при минимизации отходов)

Способ раскроя	1	2	3	4	5	6	7	8	9	10	11	12	Всего листов
Количество листов													
Всего отходов, см													

Таблица 5.24. Количество полученных заготовок

1-го вида	2-го вида	3-го вида	4-го вида

3. Найти оптимальное решение задачи, в которой при тех же исходных данных нужно *минимизировать* не отходы, а *количество используемых листов проката*.

4. Скопировать результаты решения задачи в табл. 5.25.

Таблица 5.25. 2 вариант решения (при минимизации листов проката)

Способ раскроя	1	2	3	4	5	6	7	8	9	10	11	12	Всего листов
Количество листов													
Всего отходов, см													

Таблица 5.26. Количество полученных заготовок

1-го вида	2-го вида	3-го вида	4-го вида

5. Оформить результаты решения задачи с применением цветового оформления диапазонов и ячеек. Сохранить результаты вычислений на листе с именем *РаскройЛиста* в книге Excel с именем *Поиск решения*.

Схема решения

Решение задачи представлено на рис. 5.16.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
2	Длина заготовки, см	Кол-во заготовок, выкраиваемых из одного листа при разрезе способом №												Треб. кол-во заготовок, шт	Общая длина, см	Длина листа, см		
3		1	2	3	4	5	6	7	8	9	10	11	12					
4	45	4	2	2	2	1	1	1	1	0	0	0	0	900	40500	184		
5	50	0	1	0	0	2	0	1	1	3	2	1	0	400	20000			
6	65	0	0	1	0	0	2	1	0	0	1	2	1	200	13000			
7	85	0	0	0	1	0	0	0	1	0	0	0	1	900	76500			
8	=СУММПРОИЗВ(\$A\$4:\$A\$7;\$B4:\$B7)													Итого:	150000	=СУММ(O4:O7)		
9	Отходы с листа при разрезе его 1, 2, ... 12 способом, см																	
10	Способы раскроя	1	2	3	4	5	6	7	8	9	10	11	12					
11	Полезное использование листа	180	140	155	175	145	175	160	180	150	165	180	150	Итого отходов	=СУММ(B23:M23)			
12	Отходы, см	4	44	29	9	39	9	24	4	34	19	4	34	253				
13	=\$P\$4-B11																	
14	Решение:	Количество листов, раскроенных 1, 2, ..., 12 способом																
15	Способ раскроя	1	2	3	4	5	6	7	8	9	10	11	12	Всего	=СУММ(B16:M16)			
16	Количество листов	0	0	0	0	0	0	0	0	0	0	0	0	0				
17	Количество полученных заготовок												Общие отходы с листов					
18	Длина заготовки, см	Итого												Всего отходов				
19	45	=СУММПРОИЗВ(\$B\$16:\$M\$16;\$B4:\$B4:\$M4:\$M4)												0				
20	50	0												=СУММПРОИЗВ(B12:M12;B16:M16)				
21	65	0																
22	85	0																

Рис. 5.16. Оформление задачи на листе Excel

Результаты решения

Таблица 5.27. Решение по 1 варианту

Способ раскроя	1	2	3	4	5	6	7	8	9	10	11	12	Всего
Количество листов	0	0	0	0	0	0	0	900	0	0	100	0	1000

Всего отходов:

4 000 .

Количество полученных заготовок:

900
1 000
200
900

Таблица 5.28. Решение по 2 варианту

Способ раскроя	1	2	3	4	5	6	7	8	9	10	11	12	Всего
Количество листов	0	0	0	200	0	0	0	500	0	0	0	200	900

Всего отходов:

10 600 .

Количество полученных заготовок:

900
500
200
900

Раздел 6. ВВЕДЕНИЕ В VBA ДЛЯ EXCEL

6.1. Интегрированная среда разработки VBA для Excel

6.1.1. Создание функции пользователя в VBA

В данном разделе на нескольких примерах показывается, как строятся функции пользователя. Начнем с простейшего примера – функции, вычисляющей значение по одной формуле. Затем обсудим более сложный пример, а именно расчет стоимости партии книг, когда значение функции зависит от условия. Следующий пример показывает, как создаются пользовательские функции, значениями параметров которых являются ссылки не на ячейку, а на диапазон. В последнем примере (расчет уравнения корреляции для повторяющихся данных наблюдений) показано, как конструируются, пользовательские функции, возвращающие не отдельное значение, а целый диапазон значений.

6.1.2. Редактор VBA

Пользовательские функции создаются в стандартном модуле редактора VBA. Для того чтобы попасть в редактор VBA, нужно выбрать в меню Excel команду *Сервис* → *Макрос* → *Редактор Visual Basic* или нажать комбинацию клавиш *Alt+F11*. В результате вы попадете в интегрированную среду разработки приложений VBA (рис. 6.1).

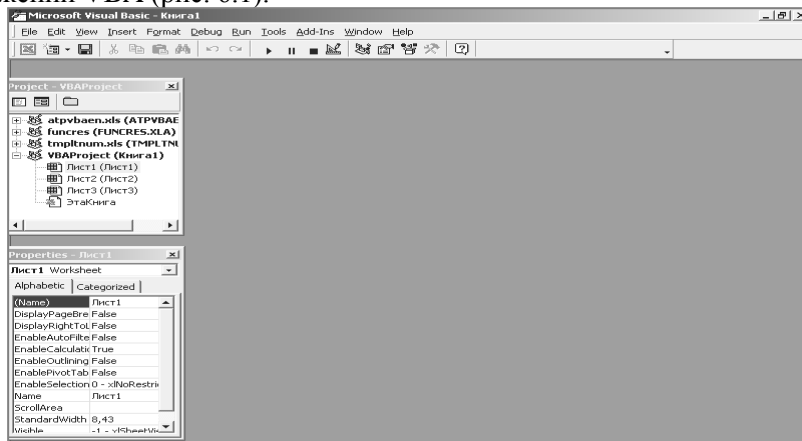


Рис. 6.1. Интегрированная среда VBA

Она имеет стандартный вид для Windows-приложений: строка меню, панель инструментов (в данном случае *Standard*) и два окна *Project – VBAProject* и *Properties*. Пока нас интересует только окно *Project – VBAProject*. В нем отображается реестр модулей и форм, входящих в создаваемый проект. Модуль, упрощенно говоря, это лист (не путать с рабочим листом), в котором набирается код. Двойным щелчком на значке модуля в окне *Project – VBAProject* можно открыть соответствующий модуль. Значок активного модуля в окне *Project – VBAProject* выделяется серым цветом. В VBA у каждого рабочего листа имеется по модулю. Кроме того, рабочая книга обладает своим модулем, и если в проекте создаются пользовательские формы, то каждая из них также имеет по модулю. В проект тоже можно добавлять модули классов для описания создаваемых пользовательских классов, но в данный момент ни один из этих модулей нас не интересует. Единственный модуль, который нужен на данный момент, – это стандартный модуль. Для того, чтобы его добавить в проект, нужно выбрать команду *Insert → Module*.

6.1.3. Структура кода функции пользователя

После того, как был создан стандартный модуль, можно перейти к написанию функции пользователя. В общем случае, функция пользователя имеет следующий вид:

```
Function ИмяФункции(СписокПараметров)
    Инструкции
End Function
```

СписокПараметров – это список параметров, от которых зависит функция. Разделителем в списке параметров является запятая.

Инструкции – это последовательность инструкций, выполняемых при нахождении значения функции. В совокупности они образуют так называемое тело функции.

Важной особенностью функции пользователя является то, что носителем возвращаемого ее значения является *ИмяФункции*. Поэтому среди инструкций должен присутствовать по крайней мере один оператор, который присваивает имени функции значение какого-либо выражения.

Допустим, досрочный выход из функции по инструкции *Exit Function*. В теле функции может располагаться несколько инструкций *Exit Function*.

6.1.4. Простейшая функция пользователя

Рассмотрим простейший демонстрационный пример построения функции пользователя, которая задается выражением $F(x) = x^2$. Для того, чтобы это сделать, достаточно в стандартном модуле (рис. 6.2) набрать следующий код:

```
Function F(x As Double) As Double
    F=x^2
End Function
```

Еще раз необходимо обратить внимание, что код пользовательской функции вводится в стандартном модуле, который добавляется в проект командой *Insert → Module*. В проекте много модулей, поэтому очень важно не перепутать их. Активный модуль в окне *Project – VBAProject* выделяется серым цветом.

Как только пользовательская функция создана, по умолчанию она попадает в раздел *Определенные пользователем* списка *Категория* окна *Мастер функций*. Найдем, например, значение этой функции при $x = 2,3$. Для этого нужно выполнить следующее:

- ввести число 2,3 в ячейку A1 рабочего листа;
- выбрать ячейку B1, в которой найти значение функции;
- выбрать команду *Вставка → Функция*;
- в первом окне *Мастера функций* выбрать функцию *F* и нажать кнопку *OK*;
- во втором окне *Мастера функций* в поле *X* ввести ссылку на ячейку A1 и нажать кнопку *OK* (рис. 6.3). Значение функции найдено.

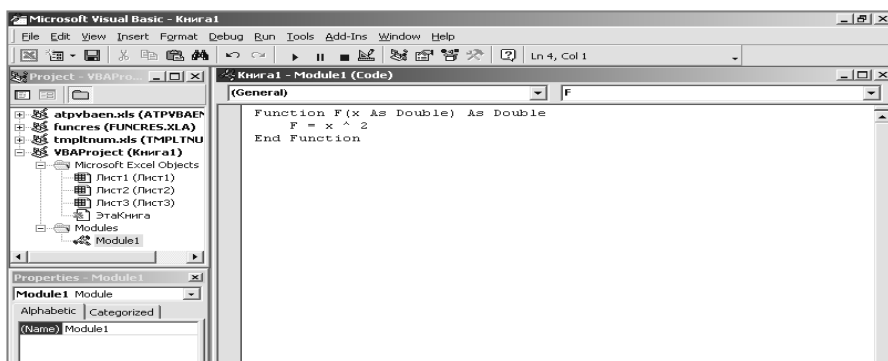


Рис. 6.2. Код пользовательской функции в модуле

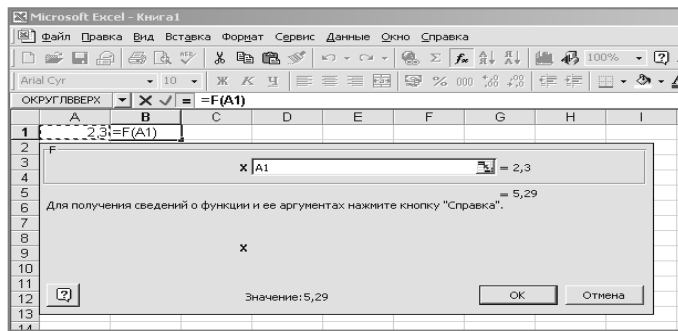


Рис. 6.3. Вычисление значения пользовательской функции при помощи Мастера функций

6.1.5. Вычисление стоимости партии продаваемых книг при помощи пользовательской функции

Естественно, что функция, приведенная в пункте 6.1.4, очень простая. Цель предыдущего примера была совершенно в другом – показать, как строятся и используются функции.

Рассмотрим более сложный пример построения функции пользователя. Допустим, что руководство издательства по оптовой продаже книг для привлечения покупателей вводит прогрессивную шкалу цен. Если продается от 100 до 200 экземпляров книги, то скидка от ее отпускной цены составляет 7%, если продается от 201 до 300 экземпляров, то скидка составляет 10%, а если свыше 300 экземпляров, то – 15%. Кроме того, для постоянных клиентов предусмотрена дополнительная скидка в размере 5%. Создадим функцию пользователя с именем *Стоимость* для расчета стоимости партии книг. Параметры этой функции назовем *ЦенаОднойКниги*, *Количество* и *Скидка*. Для параметра *Скидка* предусмотрим только два допустимых значения: 1 – для постоянных клиентов и 0 – для всех остальных. Определим пользовательскую функцию *Стоимость* следующим кодом:

```
Function Стоимость(ЦенаОднойКниги, Количество, Скидка)
    If Количество < 100 Then
        СтоимостьБезСкидки = ЦенаОднойКниги * Количество
    ElseIf Количество <= 200 Then
        СтоимостьБезСкидки = ЦенаОднойКниги * Количество * 0.93
    ElseIf Количество <= 300 Then
        СтоимостьБезСкидки = ЦенаОднойКниги * Количество * 0.9
    Else
        СтоимостьБезСкидки = ЦенаОднойКниги * Количество * 0.85
    End If
    If Скидка = 0 Then
        Стоимость = СтоимостьБезСкидки
    Else
        Стоимость = СтоимостьБезСкидки * 0.95
    End If
End Function
```

Итак, функция пользователя *Стоимость* создана. Благодаря тому, что в VBA допустимо применение русскоязычных имен, текст написанной программы ясен и прозрачен для понимания. Кроме того, это упрощает и простоту использования диалогового окна *Мастера функций* для данной функции (рис. 6.4).

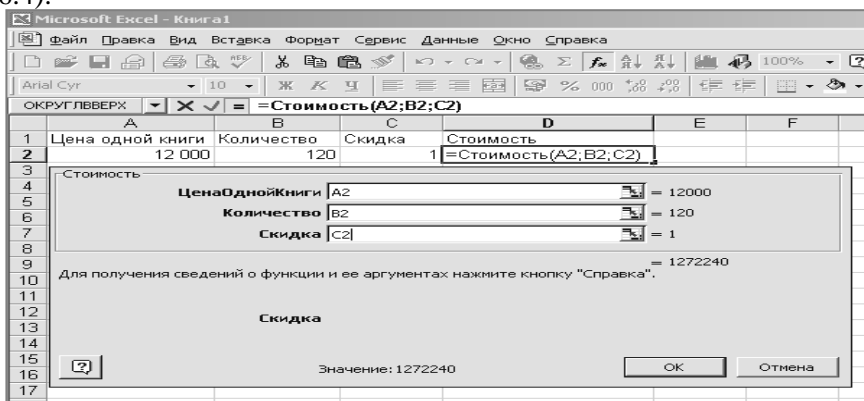


Рис. 6.4. Расчет стоимости партии книг

Названия всех параметров функции *Стоимость* в окне *Мастера функций* выводятся также на русском языке. Ясная структура диалогового окна позволяет применять функцию *Стоимость* любому пользователю, даже не владеющему VBA. Для ее использования достаточно только знать имя этой функции.

6.1.6. Использование ссылок на диапазоны в качестве параметров пользовательских функций

Пример из пункта 6.1.5 о вычислении стоимости партии книг звучит уже убедительно. VBA действительно может облегчить жизнь пользователя. Возникает только один вопрос по двум приведенным примерам. В созданных в них пользовательских функциях в качестве значений параметров были только ссылки на ячейки. Можно ли построить пользовательскую функцию, у которой в качестве значений параметров могут быть ссылки на диапазоны ячеек? Другими словами, возможно ли построить некоторый аналог функции рабочего листа СУММ, которая возвращает сумму значений из указанных диапазонов ячеек. Ответ на этот вопрос положителен. И задачу решает следующий код:

```
Function MySum(ByVal rng As Range) As Double
```

```
    Dim c As Range
```

```
    Dim s As Double
```

```
    s = 0
```

```
    For Each c In rng.Cells
```

```
        s = s + c.Value
```

```
    Next
```

```
    MySum = s
```

```
End Function
```

6.1.7. Панель инструментов «Элементы управления»

При помощи панели инструментов *Элементы управления*, которая отображается на экране выбором команды *Вид* → *Панели инструментов* → *Элементы управления*, на рабочем листе можно размещать различные элементы управления: кнопку, список, переключатель и другие (рис. 6.5).

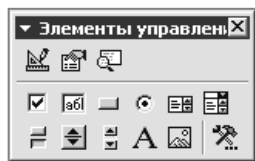


Рис. 6.5. Панель инструментов для рабочего листа

В модуле рабочего листа можно создать процедуры, обрабатывающие те или иные события, происходящие с элементами управления. Например, нажатие кнопки, выбор элемента из списка, выбор переключателя, установка флажка и другие действия, которые будут автоматически приводить к тем или иным вычислениям, построению диаграмм или смене их типа и т. д. В последующих разделах будут приведены простейшие примеры подобной автоматизации работы с рабочим листом. Первоначально ограничимся только одним элементом управления – *Кнопка* и одним событием, связанным с кнопкой, – *Click*, которое генерируется при нажатии на эту кнопку.

6.1.8. Работа с макрорекордером

Написание программы, это, в каком-то смысле, обучение компьютера делать то, что вам нужно от него. Для быстрого получения чернового варианта кода, используемого в создаваемом приложении, отлично подходит макрорекордер. *Макрорекордер* – это транслятор, создающий программу (макрос) на языке VBA, которая является результатом перевода на языке VBA действий пользователя с момента запуска макрорекордера до окончания записи макроса.

Рассмотрим, какой код записывается в макрос при активизации рабочего листа. Затем, в следующем разделе, мы используем его для построения простейшего автоматизированного содержания рабочей книги.

Итак, пусть активизирован рабочий лист *Лист1*, и мы запишем макрос, который получается при активизации рабочего листа *Лист2*.

Для активизации макрорекордера нужно выбрать команду *Сервис* → *Макрос* → *Начать запись*. Появится диалоговое окно *Запись макроса* (рис. 6.6). Это диалоговое окно позволяет задать параметры макроса.

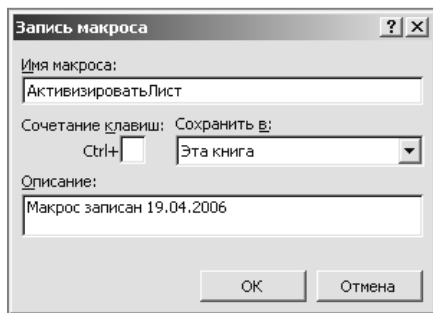


Рис. 6.6. Диалоговое окно *Запись макроса*

Поля *Имя макроса* и *Описание* используются для задания имени макроса и описания. Описание важно для многократно используемых макросов. Наша память недолговечна и, не имея подсказки в виде описания, через некоторое время будет трудно вспомнить, для чего тот или иной макрос создавался. По умолчанию макросам присваиваются имена *Макрос1*, *Макрос2* и т. д. С целью облегчения узнаваемости макроса лучше не использовать стандартное имя, а присвоить ему какое-нибудь уникальное имя, поясняющее, для чего он используется. В нашем случае, например, присвоим макросу имя *АктивизироватьЛист*. Поле *Сочетание клавиш* позволяет назначить макросу комбинацию клавиш, т. е. указать символ, который в комбинации с клавишей *Ctrl* позволит его выполнить. Назначать комбинацию клавиш макросу совсем не обязательно. Это стоит делать только для постоянно используемых макросов, для быстрого доступа к ним. Не используя комбинации клавиш, макрос можно всегда вызвать командой *Сервис → Макрос → Макросы*. Раскрывающийся список *Сохранить в* предназначен для выбора книги, в которой будет сохранен макрос. Если выбрать *Личная книга макросов*, то макрос будет сохранен в специальную скрытую книгу, в которой хранятся макросы. Эта книга всегда открыта, хотя и скрыта, и записанные в ней макросы доступны для других рабочих книг. Команда *Окно → Отобразить* позволяет отобразить личную книгу макросов. Если в раскрывающемся списке выбрать *Эта книга* (т. е. выбор, который по умолчанию предлагает компьютер), то макрос сохранится на новом листе модуля в активной рабочей книге, а если выбрать *Новая книга*, то он сохранится в новой рабочей книге.

Итак, в диалоговом окне *Запись макроса* в поле *Имя макроса* введем *АктивизироватьЛист*, а в поле *Описание* – *Активизация рабочего листа Лист2*. Нажмем кнопку *OK*. Появится плавающая панель инструментов с кнопкой *Остановить запись*. Теперь все производимые действия будут записываться до тех пор, пока не будет нажата эта кнопка или не будет выбрана команда *Сервис → Макрос → Остановить запись*.

Далее щелчком на ярлыке рабочего листа *Лист2* активизировать его и остановить запись макроса выбором соответствующей команды. Для просмотра только что записанной процедуры нужно выбрать команду *Сервис → Макрос → Макросы*, что приведет к отображению диалогового окна *Макрос* (рис. 6.7).

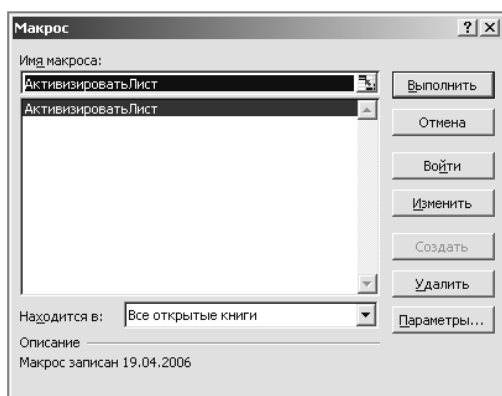


Рис. 6.7. Диалоговое окно *Макрос*

В диалоговом окне *Макрос* в списке выделим макрос и нажмем кнопку *Изменить*. На экране отобразится окно редактора VBA с активизированным стандартным модулем, в котором будет код только что записанного макроса:

```
Sub АктивизироватьЛист()
'
' АктивизироватьЛист Макрос
```

' Макрос записан 19.04.2006

```
Sheets("Лист2").Select  
End Sub
```

6.1.9. Структура кода процедуры

Итак, макрос записан, но прежде чем идти дальше, необходимо пополнить теоретический багаж. Отметим, что макрорекордер на самом деле в модуле создал процедуру. В общем случае процедура имеет следующий вид:

```
Sub ИмяПроцедуры(СписокПараметров)  
    Инструкции  
End Sub
```

СписокПараметров – это список параметров, от которых зависит функция. Разделителем в списке параметров является запятая. Список параметров, вообще говоря, может быть и пуст.

Инструкции – это последовательность инструкций, выполняемых при проведении процедуры. В совокупности они образуют так называемое «тело» процедуры.

Допустим, досрочный выход из процедуры по инструкции *Exit Sub*. В теле процедуры может располагаться несколько инструкций *Exit Sub*.

6.1.10. Процедура обработки события

Кроме обычных процедур, в VBA имеются процедуры, обрабатывающие события, связанные с тем или иным объектом. В общем случае такие процедуры имеют следующий вид:

```
[Private] Sub ИмяОбъекта_ИмяСобытия(СписокПараметров)  
    Инструкции  
End Sub
```

Ключевое слово *Private* является модификатором доступа и обозначает, что процедура видна другим процедурам только в модуле, в котором она располагается. Например, процедура обработки события активизации рабочего листа имеет вид:

```
Private Sub Worksheet_Activate()  
    Инструкции  
End Sub
```

6.1.11. Элемент управления «Кнопка» и создание процедуры обработки события

В данном разделе создадим на рабочем листе *Лист1* кнопку, нажатие на которую будет приводить к активизации рабочего листа *Лист2*. Первоначально выберем рабочий лист *Лист1*.

Прежде, чем переходить к осуществлению намеченных планов, поговорим о том, как создается кнопка на рабочем листе, что собой представляет элемент управления *Кнопка* и как пишется код обработки события *Нажатие на кнопку*.

Прежде всего переведем Excel в режим конструктора элементов управления. Для этого нужно нажать кнопку *Режим конструктора* панели инструментов *Элементы управления*. Теперь можно активизировать элемент управления *Кнопка* и нарисовать кнопку точно так же, как и при помощи любого графического редактора, в том месте рабочего листа, где нужно расположить кнопку.

Как и любой графический объект, кнопку можно рисовать при нажатой клавише *Shift*, что обеспечит ей квадратную форму, либо при нажатой клавише *Alt* – для привязки кнопки к сетке рабочего листа. У созданной кнопки при помощи маркеров изменения размеров можно задать размеры, а при помощи маркера перемещения – установить ее местоположение.

На поверхности первого созданного элемента управления *Кнопка* автоматически будет отображена надпись *CommandButton1* (рис. 6.8). Если создать второй элемент управления *Кнопка*, то на его поверхности отобразится *CommandButton2* и т. д.

Элемент управления *Кнопка* является объектом, т. е. он, как и любой объект, обладает свойствами, методами и событиями. Надпись, отображаемая на поверхности элемента управления *Кнопка*, задается значением свойства *Caption*. Кроме того, элемент управления *Кнопка* имеет свойство *Name*, которое в коде идентифицирует его как объект. В данном случае оно тоже равно *CommandButton1*.

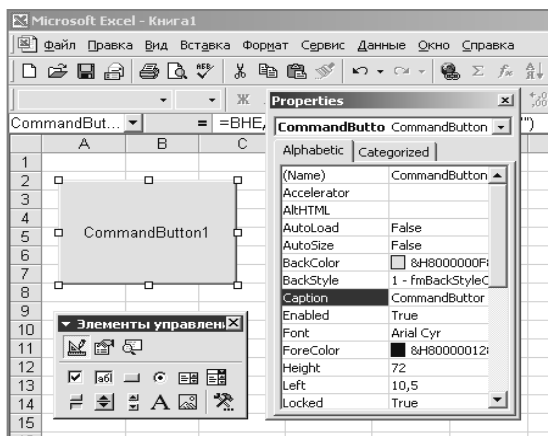


Рис. 6.8. Кнопка и окно *Properties*

Изменим значение свойства *Caption*, т. е. надпись, отображаемую на поверхности кнопки с *CommandButton1* на *Лист2* (с тем, чтобы подчеркнуть, что эта кнопка будет активизировать рабочий лист *Лист2*). Для этого надо нажать кнопку *Свойства* панели инструментов *Элементы управления*. На экране отобразится окно *Properties*. Введем в поле *Caption* этого окна значение *Лист2* и закроем окно *Properties*.

Теперь можно перейти к созданию кода процедуры, обрабатывающей событие *Нажатие на кнопку*. В результате обработки этого события должен активизироваться рабочий лист *Лист2*. Для этого дважды щелкнем на созданной кнопке (напоминаем, что все это еще происходит в режиме конструктора). В результате откроется редактор VBA с активизированным модулем рабочего листа (в данном случае, *Лист1*). Кроме того, двойной щелчок по кнопке создаст в модуле первую и последнюю инструкции процедуры обработки события *Нажатие на кнопку*:

```
Private Sub CommandButton1_Click()
End Sub
```

Вот теперь нам пригодится ранее созданный макрос. Из его кода видно, что активизацией (в данном случае, точнее сказать, выбором) рабочего листа управляет одна инструкция: *Sheets("Лист2").Select*.

Через буфер обмена скопируем эту инструкцию из макроса в процедуру обработки события *Нажатие на кнопку*. Конечно, ее можно было бы набрать и вручную. Но это довольно медленно и чревато появлением случайных опечаток, неизбежно время от времени тут и там возникающих при наборе кода. В результате процедура обработки события должна выглядеть следующим образом:

```
Private Sub CommandButton1_Click()
    Sheets("Лист2").Select
End Sub
```

Итак, процедура создана. Осталось только вернуться на рабочий лист *Лист1*. Созданная кнопка будет обрабатывать событие (нажатие на нее) только после выхода из режима конструктора. Поэтому нужно отключить режим конструктора нажатием кнопки *Режим конструктора* панели инструментов *Элементы управления*. Теперь можно и протестировать созданную кнопку, нажав на нее. Если все было правильно сделано, то это приведет к активизации рабочего листа *Лист2*.

6.1.12. Построение шаблона таблицы

Предположим, что каждый месяц составляется таблица учета расходов (рис. 6.9).

	A	B	C	D	E
1		Расходы			
2	Телефон	12 000			
3	Аренда	257 000			
4	Амортизация	21 000			
5	Страховка	35 000			
6	Заработная плата	850 000			
7	Итого	=СУММ(B2:B6)			

Рис. 6.9. Шаблон таблицы расходов

Можно облегчить подготовительную работу, автоматизировав процесс создания шаблона таблицы, в которую останется только ввести данные. Как всегда, при написании кода поможет макрорекордер.

Для построения шаблона таблицы нужно выполнить следующие действия:

1. Запустить макрорекордер на выполнение выбором команды *Сервис* → *Макрос* → *Начать запись*.
2. Выбрать ячейку B1 и ввести в нее *Расходы*.
3. Выбрать ячейку A2 и ввести в нее *Телефон*.
4. Выбрать ячейку A3 и ввести в нее *Аренда*.
5. Выбрать ячейку A4 и ввести в нее *Амортизация*.
6. Выбрать ячейку A5 и ввести в нее *Страховка*.
7. Выбрать ячейку A6 и ввести в нее *Заработная плата*.
8. Выбрать ячейку A7 и ввести в нее *Итого*.
9. Выбрать ячейку B7 и ввести в нее любым способом формулу =СУММ(B2:B6).
10. Выбрать столбец A и команду *Формат* → *Столбец* → *Автоподбор ширины*.
11. Завершить работу макрорекордера выбором команды *Сервис* → *Макрос* → *Остановить запись*.

В результате на листе стандартного модуля будет записан следующий макрос:

```
Sub Макрос1()  
' Макрос1 Макрос  
' Макрос записан 18.04.2006  
Range("B1").Select  
ActiveCell.FormulaR1C1 = "Расходы"  
Range("A2").Select  
ActiveCell.FormulaR1C1 = "Телефон"  
Range("A3").Select  
ActiveCell.FormulaR1C1 = "Аренда"  
Range("A4").Select  
ActiveCell.FormulaR1C1 = "Амортизация"  
Range("A5").Select  
ActiveCell.FormulaR1C1 = "Страховка"  
Range("A6").Select  
ActiveCell.FormulaR1C1 = "Заработная плата"  
Range("A7").Select  
ActiveCell.FormulaR1C1 = "Итого"  
Range("B7").Select  
ActiveCell.FormulaR1C1 = "=SUM(R[-5]C:R[-1]C)"  
Columns("A:A").Select  
Selection.Columns.AutoFit  
End Sub
```

Рассмотрим, как из этого непропорционально длинного (по сравнению с проделанной работой) и некачественного макроса составить разумный код для создания шаблона отчета. Первые четырнадцать строк кода макроса осуществляют попарно ввод текстовой информации в выбранные ячейки рабочего листа. Имеет смысл каждую из этих пар инструкций заменить одной.

Например, вместо

```
Range("B1").Select  
ActiveCell.FormulaR1C1 = "Расходы"
```

можно ограничиться только инструкцией

```
Range("B1").Value = "Расходы"
```

Следующие две инструкции реализуют ввод формулы =СУММ(B2:B6) в ячейку B7. При этом в макросе эта формула записана в относительном формате R1C1, что выглядит совсем не к месту:

```
Range("B7").Select  
ActiveCell.FormulaR1C1 = "=SUM(R[-5]C:R[-1]C)"
```

Разумно эти две инструкции заменить только одной, используя формат A1 и формулу локальной версии:

```
Range("B7").FormulaLocal = "=СУММ(B2:B6)"
```

Последние две инструкции макроса реализуют автоматический подбор ширины столбца A так, чтобы в нем помещались все введенные в него строки текста:

```
Columns("A:A").Select  
Selection.Columns.AutoFit
```

Целесообразно сократить код, заменив эти две инструкции одной:

```
Columns("A:A").AutoFit
```

Кроме того, для пользователя было бы удобно, чтобы имя рабочего листа совпадало с названием месяца, за который составляется отчет. Поэтому будет целесообразно в процедуре добавить инструкции, которые привели бы к отображению на экране диалогового окна, предлагающего изменить название листа. Если пользователь соглашается с этим предложением, то он вводит в поле ввода появившегося диалогового окна новое имя рабочего листа и нажимает на кнопку *ОК*, а процедура сама уж переименует лист.

Итак, код окончательного варианта процедуры составления шаблона прозрачен, он состоит всего из нескольких инструкций и, что важно, значительно более функционален, чем макрос, на основе которого он был составлен:

```
Sub ChartBuilder()  
    Dim sheetName As String  
    sheetName = InputBox("Введите имя листа")  
    If sheetName <> Empty Then  
        ActiveSheet.Name = sheetName  
    End If  
    Range("B1").Value = "Расходы"  
    Range("A2").Value = "Телефон"  
    Range("A3").Value = "Аренда"  
    Range("A4").Value = "Амортизация"  
    Range("A5").Value = "Страховка"  
    Range("A6").Value = "Заработная плата"  
    Range("A7").Value = "Итого"  
    Range("B7").FormulaLocal = "=СУММ(B2:B6)"  
    Columns("A:A").AutoFit  
End Sub
```

6.1.13. Управление диаграммой

Предположим, что надо создать годовой отчет о доходах фирмы, причем построить диаграммы дохода только за январь, за январь и февраль, с января по март и т. д. (всего 12 диаграмм). Конечно, можно построить все эти диаграммы, а можно обойтись и одной. В этом случае надо применить инструмент, позволяющий вводить заданный временной интервал, а диаграмма автоматически должна перестраиваться в соответствии с ним. Тут понадобятся элементы управления – *Список* либо *Поле со списком*.

Для создания диаграммы нужно выполнить следующие действия:

1. В диапазон A2:A13 ввести названия месяцев (рис. 6.10).
2. В диапазон B2:B13 ввести данные о доходах фирмы.
3. По этим двум диапазонам построить диаграмму.
4. На рабочем листе расположить *Поле со списком*, которое и позволит произвести выбор временного интервала.
5. При помощи окна *Properties* установить его значение *ListFillRange* равным A2:A13. Это свойство заполняет список на основе данных из указанного диапазона.

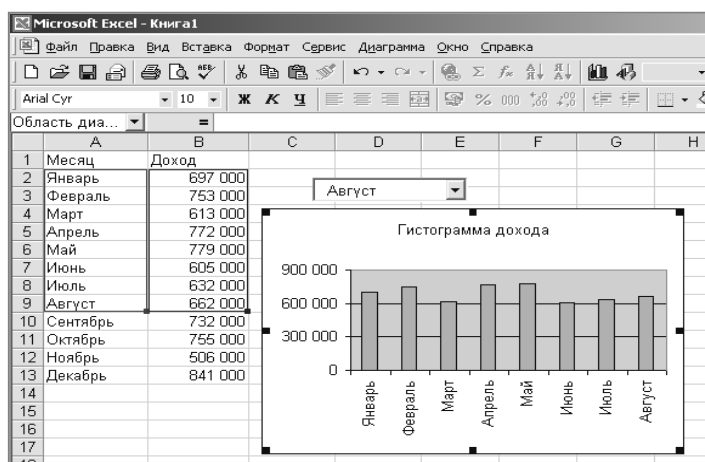


Рис. 6.10. Управление диаграммой

6. В модуле рабочего листа нужно набрать следующий код:

```
Private Sub ComboBox1_Change()  
    Dim r As Integer  
    ActiveSheet.ChartObjects(1).Activate  
    r = ComboBox1.ListIndex + 2  
    With ActiveChart  
        .SetSourceData _  
            Source:= Sheets(1).Range(Cells(2, 2), Cells(r, 2)), _  
            PlotBy:=xlColumns  
        .SeriesCollection(1).XValues = _  
            Sheets(1).Range(Cells(2, 1), Cells(r, 1))  
    End With  
End Sub
```

6.2. Создание диалоговых окон на VBA

6.2.1. Общие сведения и применение

Диалоговые окна – наиболее общее и гибкое средство организации взаимодействия пользователя с прикладной программой в Excel. Диалоговое окно представляет собой прямоугольную область экрана, где размещены элементы управления, с которыми работает пользователь. В Excel масса встроенных элементов управления, позволяющих спроектировать и реализовать интерфейс типичной офисной программы. Если же встроенных элементов управления не хватает, можно использовать в своем диалоговом окне произвольные ActiveX-объекты.

В VBA-проекте пользовательское диалоговое окно задает класс объектов. В программе можно объявлять объекты этого класса, т. е. иметь одновременно несколько его экземпляров, динамически менять свойства этих объектов и вызывать их методы, например, показывать и скрывать их.

Каждый элемент управления – это тоже объект определенного класса с предопределенным набором свойств, методов и событий. Начальные значения свойств всего диалогового окна и его отдельных элементов можно задать при проектировании окна. Затем при работе программы некоторые свойства могут меняться пользователем (например, текст в поле ввода, состояния кнопок выбора и т. п.), а некоторые – из программы. Программные изменения происходят обычно в процедурах обработки событий, автоматически запускаемых при наступлении событий, связанных с диалоговым окном и его элементами.

Чтобы диалоговое окно стало доступно пользователю, его нужно вывести на экран в ответ на некоторое действие пользователя или автоматически при некотором изменении состояния программы. Обычно диалоговые окна выводятся командами выпадающих или вспомогательных меню. Диалоговые окна могут также вызываться из других окон. Автоматически выводятся окна, в которых программа после завершения задания сообщает о полученных результатах, окна с информацией об обнаруженных ошибках и т. п.

6.2.2. Встроенные диалоговые окна

Вывод сообщений. Функция MsgBox

Диалоговое окно, создаваемое функцией *MsgBox*, служит для вывода на экран сообщения программы и получения от пользователя простой реакции на это сообщение в виде щелчка по одной из кнопок окна. В простейшем случае эта функция вызывается как процедура. Например, нижеприведенному вызову соответствует окно, показанное на рис. 6.11:

MsgBox "Сегодня на календаре" & Date

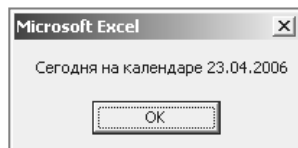


Рис. 6.11. Пример окна сообщений

После щелчка по кнопке *OK* вызов функции завершится, и выполнение программы возобновится с оператора, стоящего непосредственно за этим вызовом.

Диалоговое окно, выводимое функцией *MsgBox*, можно обогатить несколькими кнопками, чтобы пользователь мог указать с их помощью направление дальнейшего вычисления, а также присоединить к нему контекстную справку. В общем случае вызов *MsgBox* имеет вид:

MsgBox(prompt [, buttons][, title][, helpfile, context]).

Здесь параметр *prompt* (*сообщение*) – строковое выражение, значение которого выводится в окне. Его максимальная длина не должна превышать 1024 символов. Разбиение на строки в этом выражении можно производить, используя символ возврата каретки (*Chr(13)*), символ перевода строки (*Chr(10)*) или их комбинацию (*Chr(13) & Chr(10)*). Строковое выражение *title* (*заголовок*) задает заголовок окна. Если его нет, заголовком становится имя приложения. Параметры *helpfile* (*файл справки*) и *context* (*контекст*) должны присутствовать (или отсутствовать) вместе. Первый из них – имя файла, содержащего справку, которая будет выведена при нажатии клавиши *F1*, а второй – числовое выражение, задающее номер темы со справкой в этом файле.

Числовой параметр *buttons* (*кнопки*) задает виды командных кнопок, помещаемых в окне, кнопку, выбираемую по умолчанию, и модальность диалогового окна. Он получается как сумма кодов соответствующих кнопок и свойств. Коды кнопок и других свойств окна функции *MsgBox* приведены в табл. 6.1. Явно не заданный, этот параметр считается равным 0.

Таблица 6.1. Коды для формирования параметра *buttons*

Имя константы	Код	Описание
<i>Коды наборов командных кнопок</i>		
<i>vbOKOnly</i>	0	Только кнопка <i>OK</i> (Готово)
<i>vbOKCancel</i>	1	Кнопки <i>OK</i> и <i>Cancel</i> (Отказ)
<i>vbAbortRetryIgnore</i>	2	Кнопки <i>Abort</i> (Прервать), <i>Retry</i> (Продолжить) и <i>Ignore</i> (Игнорировать)
<i>vbYesNoCancel</i>	3	Кнопки <i>Yes</i> (Да), <i>No</i> (Нет) и <i>Cancel</i>
<i>vbYesNo</i>	4	Кнопки <i>Yes</i> и <i>No</i>
<i>vbRetryCancel</i>	5	Кнопки <i>Retry</i> и <i>Cancel</i>
<i>Коды пиктограмм</i>		
<i>vbCnctical</i>	16	Важное сообщение (крестик в круге)
<i>vbQuestion</i>	32	Запрос (вопросительный знак)
<i>vbExclamation</i>	48	Предупреждение (восклицательный знак)
<i>vbInformation</i>	64	Информационное сообщение (<i>i</i> в круге)
<i>Кнопка, выбранная по умолчанию</i>		
<i>vbDefaultButton1</i>	0	Первая кнопка
<i>vbDefaultButton2</i>	256	Вторая кнопка
<i>vbDefaultButton3</i>	512	Третья кнопка
<i>vbDefaultButton4</i>	768	Четвертая кнопка
<i>Модальность диалога</i>		
<i>vbApplicationModal</i>	0	Модален относительно приложения
<i>vbSystemModal</i>	4096	Модален относительно системы

Диалоговое окно модально относительно приложения, если для продолжения работы в текущем приложении пользователь должен закончить работу в этом окне (ответить на сообщение). Модальность относительно системы означает, что все приложения будут приостановлены до завершения работы в данном окне.

Может показаться странным, что константа *vbDefaultButton4* указывает на четвертую кнопку, хотя максимальное количество кнопок, определяемых кодами от 0 до 5, – три. Дело в том, что некоторые приложения (например, Excel) при наличии параметров *helpfile* и *context* могут автоматически добавить в окно кнопку справки *Help*, вызывающую окно справки.

Результирующее значение параметра *buttons* получается как сумма кодов по одному из каждой группы. Например, значение $36 = 4 + 32 = \textit{vbYesNo} + \textit{vbQuestion}$ означает, что в окне будет пиктограмма с вопросительным знаком и две кнопки *Yes* и *No*, причем первая из них будет считаться выбранной, если пользователь щелкнет по ней кнопкой мыши или нажмет клавишу *Enter*.

Как узнать, какую кнопку выбрал пользователь? Присвоить значение функции *MsgBox* некоторой целочисленной переменной и проверить его. Возвращаемые *MsgBox* значения приведены в табл. 6.2.

Таблица 6.2. Коды, возвращаемые функцией *MsgBox*

Имя константы	Значение	Нажатая кнопка
<i>vbOK</i>	1	<i>OK</i>
<i>vbCancel</i>	2	<i>Cancel</i>
<i>vbAbort</i>	3	<i>Abort</i>

Имя константы	Значение	Нажатая кнопка
vbRetry	4	<i>Retry</i>
vbIgnore	5	<i>Ignore</i>
vbYes	6	<i>Yes</i>
vbNo	7	<i>No</i>

Если в окне есть кнопка *Cancel*, нажатие клавиши *Esc* завершает работу в окне и возвращается тот же код *vbCancel*, что и при выборе кнопки *Cancel*.

В следующем примере создается окно с сообщением об ошибке и вопросом о продолжении вычисления. В нем пиктограмма важного сообщения и кнопки *Yes* и *No*, причем вторая – кнопка, выбираемая по умолчанию. Файл со справкой называется *ERRORS.HLP*, а номер контекстной справки в этом файле равен 200.

Последовательность создания окна следующая:

```

Dim Msg As String, MyString As String
Dim Btns As Integer
Dim Title As String, Help As String
Dim NmbCont As Integer, Result As Integer
Title = "Вы ввели неверные данные !" ' заголовок окна
Msg = "Будем продолжать работу ?" ' сообщение
' кнопки и свойства
Btns = vbYesNo + vbCritical + vbDefaultButton2
Help = "ERRORS.HLP" ' имя файла со справкой
NmbCont = 200 ' номер темы
' Вывод сообщения.
Result = MsgBox(Msg, Btns, Title, Help, NmbCont)
If Result = vbYes Then ' пользователь нажал кнопку Yes
    MyString = "Да" ' действия по продолжению работы
Else ' пользователь нажал кнопку No
    MyString = "Нет" ' действия по прекращению работы
End If

```

При этом вызове на экране появится окно, представленное на рис. 6.12.

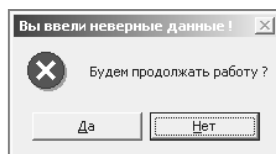


Рис. 6.12. Сообщение о неверных данных

Окно ввода данных. Функция *InputBox*

Окно, создаваемое функцией *InputBox*, предназначено для ввода строки и содержит однострочное поле ввода, сообщение и кнопки *OK* и *Cancel*. При выборе кнопки *OK* (или нажатии клавиши *Enter*) строка, введенная пользователем в поле ввода, передается в программу, щелчок кнопки *Cancel* (нажатие клавиши *Esc*) означает отказ от ввода – *InputBox* возвращает пустую строку.

Синтаксис вызова функции *InputBox*:

```

InputBox(prompt [, title] [,default] [, xpos] [, ypos]
[, helpfile, context])

```

Здесь параметры *prompt* (сообщение), *title* (заголовок), *helpfile* и *context* имеют тот же смысл, что и для функции *MsgBox*. Числовое выражение *xpos* задает расстояние по горизонтали левой границы окна от левой границы экрана. Если его нет, окно центрируется по горизонтали. *Ypos* – расстояние верхней границы окна от верхней границы экрана. Если его нет, окно центрируется по вертикали.

В следующем примере в окно функции *InputBox* пользователь должен ввести дату рождения. Далее она проверяется. Если она в интервале от 01.01.1900 до 01.01.1985, запрашивается имя пользователя еще одним вызовом функции *InputBox*, иначе выдаются информационные сообщения с помощью функции *MsgBox*. Так как возвращаемое при первом вызове функции *InputBox* значение – строка, для его преобразования в дату применяется функция *DateValue* (для преобразования вводимых строк в числовые значения используется функция *Val*). На рисунках 6.13 – 6.15 представлен один из возможных диалогов.

Процесс введения даты рождения и ее проверки следующий:

```

Sub exInput()
    Dim Msg As String, Title As String, Name As String
    Dim Birthday As Date
    Msg = "Введите, пожалуйста, " & Chr(13) & "дату своего рождения"
    Title = "Окно для ввода даты"
    ' Вывод окна для ввода даты в позицию (1500,1000):
    Birthday = DateValue(InputBox(Msg, Title, , 1500, 1000))
    ' Проверка даты:
    If Birthday < #01/01/1900# Then
        MsgBox "Вы выглядите гораздо моложе! ", VbExclamation
    ElseIf Birthday > #01/01/1981# Then
        MsgBox "Эта программа не для детей до 16! ", VbExclamation
    Else
        ' Дата корректна. Запрос имени пользователя
        Name = InputBox("Здравствуйте! Как Вас зовут?", "Знакомство")
    End If
    MsgBox "Добрый день, " & Name, VbExclamation, "Проверка ввода"
End Sub

```

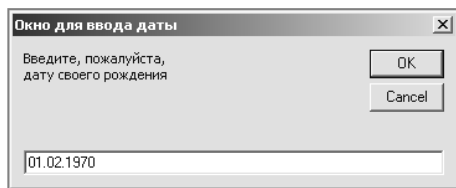


Рис. 6.13. Окно для ввода даты

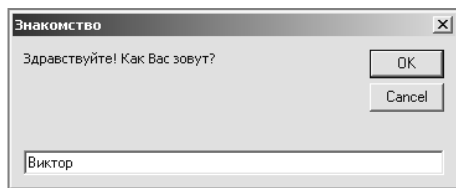


Рис. 6.14. Окно для ввода имени

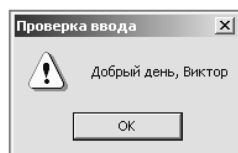


Рис. 6.15. Результат обработки ввода

6.2.3. Создание пользовательских диалоговых окон

Новые пользовательские диалоговые окна проектируются в редакторе VBA. В процессе создания диалогового окна вначале определяется его видимое изображение на экране (форма с управляющими элементами), затем задаются их свойства и разрабатываются процедуры, обрабатывающие события, связанные с формой и ее элементами управления.

Для создания новой формы нужно в меню редактора VBA выбрать команду меню *Insert* → *UserForm*. На экране появится новое окно с пустой формой и расположенной рядом панелью инструментов *Элементы управления*. Нужно щелкнуть по этому окну правой кнопкой мыши и выбрать в появившемся контекстном меню команду *Properties* (*Свойства*). На экране появится окно *Properties UserForm* со списком свойств нового окна. Если выбрать вкладку *Categorized* (*По категориям*), будет показан список свойств, сгруппированных по категориям. В частности, в категории *Appearance* (*Вид*) можно задать цвета окна и его границ и установить нужный заголовок окна в свойстве *Caption*. Свойства из других групп позволяют установить шрифт, используемый в окне, положение и размер окна и другие.

Чтобы подробнее узнать о свойстве, надо щелкнуть по нему и нажать клавишу *F1*. Затем нужно расположить в окне управляющие элементы. Они выбираются на панели инструментов *Элементы управления*. Необходимо перетащить нужный элемент с панели элементов на предназначенное место в окно формы. Отпустить кнопку мыши. Вокруг элемента появится прямоугольная рамка с выделенными точками, за которые ее можно перемещать по экрану, чтобы точнее расположить элемент управления в диалоговом окне.

и установить его размеры. Уточнить положение элемента в окне можно командами меню *Format*. Порядок обхода управляющих элементов формы устанавливает команда *TabOrder* (*Последовательность перехода*) из меню *View* или контекстного меню формы. Чтобы данный управляющий элемент никогда не попадал в фокус, нужно щелкнуть по нему правой кнопкой мыши, выбрать в появившемся контекстном меню команду *Properties*, а затем в появившемся списке установить для свойства *TabStop* значение *False*. Отметим, что для текстовых надписей (элементов вида *Label*) это значение устанавливается по умолчанию. В окне свойств элемента можно установить при проектировании и другие свойства элемента управления. Списки свойств появляются в этом окне на вкладках *Alphabetic* (*По алфавиту*) и *Categorized* (*По категориям*). На первой – в алфавитном порядке, на второй – по категориям. Устанавливать значение свойства можно на любой из них.

VBA автоматически именует новые устройства управления, образуя имя из типа устройства и его порядкового номера среди устройств данного типа. Например, у второй командной кнопки по умолчанию будет имя *CommandButton2*. Элементу можно присвоить новое имя, установив его в качестве значения свойства *Name*. Важно помнить, что имя идентифицирует объект (элемент управления или форму) в программе, поэтому изменение имени должно повлечь изменения в текстах процедур, его использующих. Не надо путать имя объекта *Name* со свойством *Caption* (*заголовок, надпись*), имеющимся у многих элементов управления. По умолчанию при создании элемента система присваивает этому свойству то же значение, что и имени *Name*. Изменение свойства *Caption* отражается на «внешности» элемента (надписи на экране), а на доступ к объекту из программы не влияет.

Можно одновременно изменить значение свойства для нескольких выделенных элементов управления, щелкнув по ним при нажатой клавише *Ctrl*, установив нужное значение свойства для одного из этих элементов. Если элемент или группа элементов будут использоваться в нескольких диалоговых окнах, их можно после настройки в первом из этих окон перетящить на панель элементов и оттуда уже в готовом виде перетаскивать на остальные окна.

6.2.4. Создание вкладок в диалоговых окнах

Элемент *MultiPage* (набор вкладок) позволяет создавать вкладки с разными элементами управления, а элемент *TabStrip* (полоса вкладок) создает вкладки с одинаковыми наборами элементов управления. Чтобы создать новый набор вкладок, нужно перетящить на форму элемент *MultiPage*. Вначале будут созданы две вкладки, которым можно дать нужные заголовки, сделав их значениями свойства *Caption* в списке свойств. На рис. 6.16 эти вкладки получили имена *Ссуды* и *Ренты*.

Для того, чтобы добавить следующую вкладку, надо щелкнуть правой кнопкой мыши правее имени последней вкладки и в появившемся контекстном меню выбрать команду *New Page*. Это меню позволяет также переименовывать, перемещать и удалять вкладки. Повторив операцию создания вкладки несколько раз, можно разместить в окне необходимое число вкладок. Затем на каждую из них нужно перетящить элементы управления.

Аналогично создаются вкладки элементом *TabStrip*. Отличие в том, что каждый помещаемый на одну вкладку элемент управления одновременно попадает на все вкладки набора.

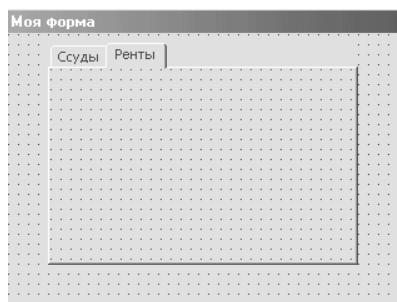


Рис. 6.16. Создание вкладок

6.2.5. Добавление дополнительных элементов управления

На панели элементов управления находятся кнопки не всех элементов, доступных в Excel. Доступ к дополнительным элементам управления можно получить, щелкнув правой кнопкой мыши по панели элементов и выбрав в появившемся контекстном меню команду *Additional Controls* (*Дополнительные элементы*). На экране появится одноименное окно со списком *Available Controls* (*доступных элементов управления*).

В этом списке отмечены элементы, находящиеся в данный момент на панели инструментов. Кроме этих, есть более 40 дополнительных элементов управления. Отметив нужные элементы в списке и выйдя из этого окна, можно увидеть их на панели инструментов и использовать при создании своих диалоговых окон.

6.2.6. Создание диалогового окна

Создадим диалоговое окно *Цветная форма*, в котором будут расположены изображение (*Image*), поле ввода (*TextBox*) и командная кнопка (*CommandButton*). Для этого нужно произвести следующие действия:

1. Выполнить команду меню *Insert* → *UserForm* для создания нового диалогового окна.
2. Щелкнув правой кнопкой появившееся окно *UserForm1*, выбрать в контекстном меню команду *Properties*. Ввести новое имя *ColorForm1* в правом столбце строки со свойством *Name*, в качестве значения свойства *Caption* ввести заголовок окна – *Цветная форма*. Щелкнуть по правому столбцу в строке со свойством *BackColor* (*Цвет фона*), затем – по появившемуся справа значку списка. В таблице появившегося окна щелчком выбрать понравившийся цвет. Окно «перекрасится». Закрыть окно свойств.
3. Перетянуть с панели инструментов на окно элемент *Image* (*Рисунок*). Щелкнуть по нему правой кнопкой мыши и снова выбрать команду *Properties*. В появившемся окне со списком свойств найти строку со свойством *Picture* (*Рисунок*). Чтобы перейти к поиску нужного файла с картинкой, надо щелкнуть по кнопке с многоточием (...) справа в этой строке. В диалоговом окне *Load Picture* (*Загрузка рисунка*) выбрать файл с картинкой и щелкнуть по кнопке *OK*.
4. Перетянуть на диалоговое окно элемент управления *TextBox* (*Поле ввода*). Как и на шаге 3, щелкнув по нему правой кнопкой, вызвать окно *Properties*. Ввести в качестве значения свойства *Value* название выбранной картинки – оно появится в поле ввода. Затем выбрать свойство *Font*. Справа появится кнопка с многоточием. Щелкнуть по ней и в появившемся списке шрифтов выбрать шрифт для поля.
5. Перетянуть на диалоговое окно командную кнопку (*Command Button*). Вызвать окно *Properties* для этой кнопки и изменить в нем значение свойства *Caption* на *Нажмите здесь* – текст станет именем кнопки. Изменить имя кнопки *Name* на *cmdClickMe*. По этому имени к кнопке будут обращаться процедуры и методы в программе. Ввести справку *Командная кнопка* как значение свойства *ControlTipText* – надпись будет появляться на экране под кнопкой всякий раз, когда на ней окажется указатель. Установить для кнопки ключ быстрого выбора, найдя в списке свойств строку со свойством *Accelerator* (*Ускоритель*) и введя в поле справа букву *H* – в имени кнопки первый символ *H* будет подчеркнут и нажатие клавиш *Alt+H* будет эквивалентно выбору кнопки. Установить также подходящий шрифт с помощью свойства *Font*, как и для поля ввода.
6. В меню *Run* (*Запуск*) выбрать команду *Run Sub* → *UserForm*. На экране приложения появится спроектированное диалоговое окно. Один из вариантов проектирования диалогового окна приведен на рис. 6.17.
7. Щелкнуть по кнопке *Закрыть* на заголовке диалогового окна, чтобы выйти из него.



Рис. 6.17. Пример проектирования диалогового окна

6.2.7. Разработка процедур, обрабатывающих события диалогового окна и его устройств

Процедуры, обрабатывающие события, связанные с диалоговыми окнами и элементами управления, – это частный случай процедур, обрабатывающих события объектов Excel. С диалоговым окном (формой) и с элементом управления каждого типа связан заранее определенный набор событий, возникающих для этого объекта по инициативе пользователя или программы. Например, одно из связанных с формой событий *Initialize* (*Инициализация*) возникает при загрузке формы, а связанное с командной кнопкой (*Command Button*) событие *Click* (*Щелчок*) возникает, когда пользователь щелкает по этой кнопке (или нажимает клавишу *Enter*, если эта кнопка определена как нажимаемая по умолчанию). Процедуры обработки событий позволяют описать поведение диалогового окна или отдельного элемента управления при наступлении того или иного события.

Чтобы перейти в окно редактора для ввода текста процедуры обработки события, нужно дважды щелкнуть кнопкой мыши по объекту, к которому оно относится (элемент управления или свободную от элементов часть формы). Появится окно *Code* (*Код*). В раскрывающемся списке слева сверху перечислены

все объекты формы, при выборе объекта в этом списке справа сверху появляются события, связанные с данным объектом. На рис. 6.18 показан список событий для командной кнопки.

События, для обработки которых процедуры уже написаны, выделены полужирным начертанием. Чтобы создать новую процедуру или отредактировать имеющуюся, нужно выбрать щелчком мыши в списке соответствующее событие. На экране появится шаблон новой процедуры или текст написанной. Имя такой процедуры состоит из имени устройства или формы, за которым после подчеркивания идет имя события. Так, процедура, обрабатывающая событие *Click* для командной кнопки *CommandButton1*, называется *CommandButton1_Click*. Из-за такого способа именования процедур надо быть осторожным при переименовании элементов управления. Если Вы сначала определили для элемента процедуры обработки события, а затем переименовали его (изменив свойство *Name*), никаких процедур обработки события у элемента с новым именем не будет (их придется перенести с помощью редактора).

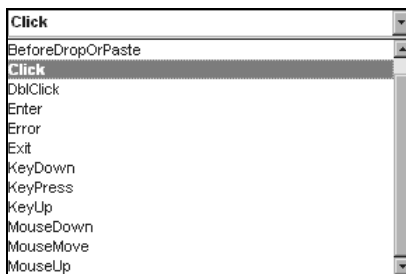


Рис. 6.18. События элемента *CommandButton*

Процедура обработки события *Click* для командной кнопки *cmdClickMe* диалогового окна *Цветная форма* следующая:

```
Private Sub cmdClickMe_Click()  
    MsgBox "Поздравляю"  
    ColorForm1.TextBox1.Value = "Цветок"  
End Sub
```

Эта процедура автоматически вызывается при выборе кнопки *Нажмите здесь* и выводит на экран поздравление, а затем заменяет название картинки в поле ввода.

Количество свойств, методов и событий, связанных с диалоговыми окнами и элементами управления, чрезвычайно велико.

6.2.8. Вызов собственного диалогового окна

Диалоговое окно нуждается в отладке. В интегрированной среде VBA можно видеть его изображение, но управляющие элементы находятся в «замороженном» состоянии, и их реакция на действия пользователя – это реакция среды, а не самих элементов. Диалоговое окно запускается на исполнение командой запуска процедуры или формы *Run → Run Sub → UserForm* интегрированной среды VBA или нажатием клавиши *F5*. Ваше диалоговое окно появится в рабочем состоянии на основной панели Excel.

Вызывается диалоговое окно в прикладной программе методом *Show* (*Показать*). Он позволяет вывести окно на экран в качестве реакции на выбор соответствующей команды меню или командной кнопки в другом окне и т. п. Например, диалоговое окно *MyForm* выводится так: *MyForm.Show*.

6.2.9. Установка начальных значений свойств элементов управления

Начальное значение (или значение по умолчанию) свойства элемента управления можно установить в процедуре, обрабатывающей событие *Initialize* (*Инициализация*) диалогового окна, содержащего данный элемент. Это событие возникает всякий раз при выводе диалогового окна на экран и запускает на выполнение процедуру без аргументов с именем *ИмяОкна_Initialize*. Поэтому установленные в ней значения свойств всегда будут начальными. Это не значит, что они должны быть постоянными. Значения могут зависеть от данных в ячейках рабочего листа Excel, от их форматов и т. п.

Для формирования диалогового окна с процедурой инициализации нужно выполнить следующие действия:

1. Создать новое диалоговое окно и разместить в нем статический текст *Label* (*Метка*), поле ввода *TextBox*, окно списка *ListBox* и кнопку независимого выбора *CheckBox* (*флажок*).

2. Последовательно вызывая для каждого из этих элементов окно *Properties*, изменить их имена: свойство *Name* метки *Label* – на *CurrDate*, поля ввода *TextBox* – на *Mytxt*, списка *List Box* – на *LstColors* и флажка *CheckBox* – на *chkGood*. Вызвать список свойств для всего диалогового окна *UserForm* и задать в качестве имени окна *frmInit*. Эти изменения внутренних имен элементов и окна не скажутся на его изоб-

ражении в интегрированной среде.

3. Дважды щелкнуть кнопкой мыши в диалоговом окне, чтобы вызвать окно *Code*. Установить в списке объектов слева сверху *UserForm* (по умолчанию этот объект и будет выбран при вызове окна кода) и выбрать команду *Initialize* в списке процедур справа сверху. На экране появится шаблон процедуры инициализации диалогового окна *UserForm_Initialize*. Ввести текст следующей процедуры:

```
Private Sub UserForm_Initialize()  
    With frmInit  
        'заголовок диалогового окна  
        .Caption = "Окно после инициализации"  
        'метка - текущая дата  
        .CurrDate.Value = "Сегодня " & Format(Date, "dd/mm/yy")  
        'начальный текст в поле ввода  
        .Mytxt.Text = "Любимый цвет-"  
        .chkGood.Value = True 'включение флажка  
        .chkGood.Caption = " Хороший день" ' заголовок флажка  
        With .LstColors 'задание элементов списка:  
            .AddItem "белый"  
            .AddItem "черный"  
            .AddItem "синий"  
            .AddItem "красный"  
            .AddItem "зеленый"  
            .AddItem "желтый"  
            .AddItem "голубой"  
            'выбор 7-го элемента в списке ("голубой")  
            .ListIndex = 6  
        End With  
    End With  
End Sub
```

Нумерация элементов массивов и коллекций, связанных с формами, начинается с 0. Поэтому, чтобы выделить при выводе окна последний (седьмой) элемент в списке цветов, свойству *ListIndex* присвоено значение 6.

4. Запустить диалоговое окно в отладчике. Его вид приведен на рис. 6.19.

5. Щелкнув кнопку *Close* (*Закреть*) на строке заголовка окна, нужно возвратиться в режим проектирования.

Для упрощения написания кода процедур VBA позволяет в качестве имени текущего диалогового окна применять ключевое слово *Me*. В частности, в предыдущем примере вместо оператора *With frmInit* можно было использовать *With Me*.

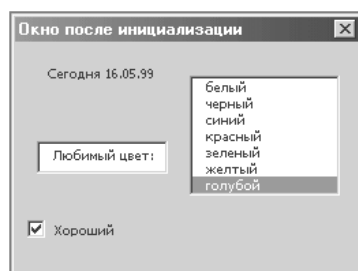


Рис. 6.19. Окно после инициализации

6.2.10. Модификация управляющих элементов во время работы

Устанавливать новые значения свойств элементов управления можно не только при инициализации диалогового окна, но и во время работы пользователя (и программы) в нем. В любой процедуре обработки события, связанного со всем диалоговым окном или его отдельным элементом, можно присвоить значение свойству или вызвать метод любого управляющего элемента. Например, эти команды изменяют текст в поле ввода и снимают флажок в окне предыдущего примера (см. п. 6.2.9):

```
'новый текст в поле  
With Me  
    .Mytxt.Text = "Нелюбимый цвет:"
```

```

'отключение флажка
.chkGood.Value = False
End With

```

Благодаря таким изменениям, программа может гибко реагировать на действия пользователя в диалоговом окне и использовать одно диалоговое окно для решения нескольких связанных или однотипных задач.

6.2.11. Управление доступом к элементу

У всех элементов управления есть свойство *Enabled* (*Доступен*), позволяющее сделать этот элемент активным – поместить его в фокус. Если свойству *Enabled* присвоить значение *False*, элемент управления становится недоступен и не может попасть в фокус при нажатии клавиши *Tab*. Когда динамически менять доступность элементов управления, зависит от задач, решаемых в диалоговом окне. Типичный случай – запрет на доступ к окну поля ввода после того, как программа поместила в него данные, которые не должны далее изменяться пользователем (например, текст справки, которую можно только прочесть и распечатать). Другой случай – отключение (включение) доступа к группам переключателей или кнопок зависимо от выбора (*OptionButton*) или отдельным таким кнопкам в зависимости от состояния флажка (*CheckBox*). При этом действия по изменению доступа надо проводить в процедуре, обрабатывающей событие *Change* для флажка. Она должна автоматически вызываться, когда этот флажок меняет свое состояние, т. е. становится неотмеченным или отмеченным. Для изменения доступа нужно выполнить следующие действия:

1. Создать новое диалоговое окно (*UserForm*), включить в него один флажок (*CheckBox*), управляющий элементом *Frame* (*Рамка*) и поместить в рамку четыре переключателя (*OptionButton*). Заголовки управляющих элементов (значения свойства *Caption*) задаются как на рис. 6.20.

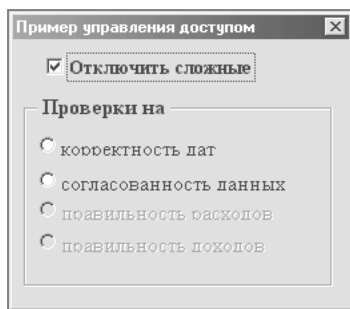


Рис. 6.20. Окно после отключения двух опций

2. Дважды щелкнуть кнопкой мыши по флажку, чтобы вывести окно с кодом процедур, выбрать в списке объектов слева сверху *CheckBox1* (это имя присвоила флажку среда VBA), затем в списке процедур обработки событий справа сверху щелкнуть кнопкой мыши по событию *Change*. Дополнить появившуюся заготовку процедуры *CheckBox1_Change*:

```

Private Sub CheckBox1_Change()
With Me
    If .CheckBox1.Value = True Then
        .OptionButton3.Enabled = False
        .OptionButton4.Enabled = False
    Else
        .OptionButton3.Enabled = True
        .OptionButton4.Enabled = True
    End If
End With
End Sub

```

3. Запустить диалоговое окно и установить флажок *Отключить сложные*. Вид получившегося окна приведен на рис. 6.20.

6.2.12. Перемещение фокуса на элемент управления

Обычно фокус перемещается по элементам управления диалогового окна в прямом направлении при нажатии пользователем клавиши *Tab*, а в обратном – комбинации *Shift + Tab*. Порядок перемещения задается на стадии проектирования окна. В процессе работы программы можно переместить фокус методом *SetFocus* (*Поместить в фокус*) элемента, который должен оказаться в фокусе.

Допустим, поле ввода *Mytxt* диалогового окна служит для ввода обязательной информации, а при вы-

боре кнопки *InputData* должны вводиться данные из этого поля. Если при выборе кнопки *InputData* данных в поле ввода нет, должен раздаться звуковой сигнал, а фокус возвратиться в поле ввода. Для реализации этого сценария в процедуру обработки события *Click* командной кнопки включим следующий фрагмент:

```
Private Sub InputData_Click()  
    If Me.Mytxt.Value = "" Then      ' данных нет  
        Beep  
        Me.Mytxt.SetFocus           ' установка фокуса на поле  
        Exit Sub  
    Else                            ' обработка введенных данных  
        Me.Mytxt.Text "Данные не пусты"  
    End If  
End Sub
```

6.2.13. Изменение размеров диалогового окна

Из процедур обработки событий можно менять свойства не только отдельных элементов управления, но и всего окна. В качестве примера рассмотрим динамическое изменение его размеров. Оно может понадобиться, когда на стадии проектирования были помещены на окно элементы управления, которые при запуске окна не нужны, но могут понадобиться в процессе работы пользователя в нем. Пусть в диалоговом окне *Myform1* есть командная кнопка *GetData* со значением свойства *Caption* равным *Получить данные*, а в его нижней части расположены поля ввода *Доходы* и *Расходы*, где эти данные будут показаны. Предположим, высота всего окна (значение свойства *Height*) равна 150, командная кнопка *GetData* находится на расстоянии 10 от верхней границы окна (10 – значение свойства *Top*), а поля ввода находятся внизу окна на расстоянии 110 от его верхнего края. Тогда, чтобы ненужные пока поля ввода *Доходы* и *Расходы* не попали в диалоговое окно, процедура инициализации должна установить высоту окна равной 90:

```
Private Sub UserForm1_Initialize()  
    Me.Height = 90  
End Sub
```

После нажатия кнопки *Получить данные* и завершения вычислений можно восстановить высоту диалогового окна, и поля *Доходы* и *Расходы* станут видны. Эту операцию будет естественным задать в процедуре, обрабатывающей событие *Click* для командной кнопки:

```
Private Sub GetData_Click()      ' Вычисление доходов и расходов ...  
    Me.Доходы = 1500  
    Me.Расходы = 1000           ' для примера  
    Me.Height = 150             ' восстановление высоты окна  
End Sub
```

6.2.14. Проверка корректности данных

Одна из важных функций диалоговых окон – прием и передача данных, введенных пользователем в программу, для хранения и дальнейшей обработки. Основной вид элементов управления, предназначенных для ввода, – это поля ввода (*TextBox*). Для предотвращения ошибок, связанных с неверным типом введенных данных или нарушением некоторых условий, которым эти данные должны удовлетворять, следует проверять их корректность до выхода из диалогового окна. Для такой проверки подходят две процедуры: обработка выхода из поля ввода, связанная с событием *Exit*, и обработка события *Click* для командной кнопки, вызывающей закрытие диалогового окна. Пример проверки при выходе из окна будет рассмотрен выше при обсуждении автоматического перемещения фокуса.

Рассмотрим далее, как оставить поле ввода в фокусе, если введенные данные некорректны. Предположим, в диалоговом окне есть поле ввода *InputData*, куда пользователь должен ввести целочисленное значение – температуру воздуха в градусах по Цельсию. Пусть глобальные константы *MinT* и *MaxT* задают минимально и максимально допустимую температуру. Следующая процедура запускается при попытке покинуть поле ввода, проверяет корректность введенных данных, в случае ошибки выдает сообщение об этом и оставляет фокус на поле ввода:

```
Private Sub InputData_Exit (ByVal Cancel As MSForms.ReturnBoolean)  
    If Not IsNumeric(InputData.Text) Then  
        MsgBox "Ошибка ввода. Не числовые данные."  
        Cancel = True  
    End Sub  
End If
```

```

If (MinT > InputData.Text) Or (InputData.Text > MaxT) Then
    MsgBox "Ошибка ввода. Выход за границы."
    Cancel = True
End If
End Sub

```

Присвоение аргументу *Cancel* значения *True* оставляет фокус на текущем элементе (в данном случае – в поле ввода *InputData*).

6.2.15. Сохранение данных перед выходом из диалогового окна

Одна из основных целей диалоговых окон – передавать от пользователя программе информацию о данных, введенных в поля ввода, элементах, выбранных в списках, включенных или выключенных флажках и т. п. Если не перенести информацию в программу до выгрузки диалогового окна, она будет утеряна, и при обращении к значениям свойств всех элементов управления будут возвращаться их начальные значения, установленные по умолчанию. Поэтому нужные программе данные следует сохранять до окончания работы диалогового окна. Для этого можно использовать рабочую память самого приложения (ячейки рабочих листов в Excel), внешние файлы или глобальные переменные. Данные целесообразно сохранять в процедурах обработки события *Click* для командных кнопок, вызывающих завершение работы диалогового окна.

Допустим, в диалоговом окне *Myform2* есть поле ввода *txtName*, куда пользователь вводит свою фамилию, список *LstColors*, где он выбирает любимый цвет, флажок *chOpt* (*Caption* равно *Полная оптимизация*) и две командных кнопки: *cmdCancel* (*Caption* равно *Отказ*) и *cmdOK* (*Caption* равно *Готово*). Сохранить информацию, указанную в этом окне, можно следующим образом:

```

' Объявления глобальных переменных (на уровне модуля Module)
Public strName As String           ' фамилия
Public strColor As String          ' любимый цвет
Public varOptim As Variant         ' признак полной оптимизации
Public btnCancelled As Boolean     ' признак отказа от ввода

' Процедуры диалогового окна
Private Sub UserForm_Initialize()
    btnCancelled = True           ' инициализация признака отказа
    With Me.LstColors             ' задание элементов списка:
        .AddItem "белый"
        .AddItem "черный"
        .AddItem "синий"
        .AddItem "красный"
        .AddItem "зеленый"
        .AddItem "желтый"
        .AddItem "голубой"
        .ListIndex = 1           ' выбор элемента списка "черный"
    End With
End Sub

Private Sub cmdCancel_Click()      ' нажата кнопка "Отказ"
    btnCancelled = True
    Unload Me
End Sub

Private Sub cmdOK_Click()         ' нажата кнопка "Готово"
    With Me                       ' Сохранение данных:
        strName = .txtName.Text
        strColor = .LstColors.Value
        varOptim = chOpt.Value
    End With
    btnCancelled = False
    Unload Me
End Sub

```

Ниже приведен пример процедуры, использующей окно *Myform*:

```

Sub exPrMyform2()
    Dim Optimization As String
    Myform2.Show                  ' вывод окна
    If btnCancelled = True Then   ' выход по кнопке "Отказ"
        MsgBox "Отказ от ввода данных", vbExclamation
    End If
End Sub

```

```

Else                                ' выход по кнопке "Готово"
    If varOptim = 0 Then
        Optimization = "Без оптимизации"
    Else
        Optimization = "С полной оптимизацией"
    End If
    MsgBox "Фамилия " & strName & _
        ". Любимый цвет " & strColor & Chr(13) & Optimization
End If
End Sub

```

На рис. 6.21 показано это окно с заполненным полем *Фамилия* и установленным флажком. После выхода из окна по нажатию кнопки *Готово* на экране появляется окно с сообщением, приведенное на рис. 6.22.

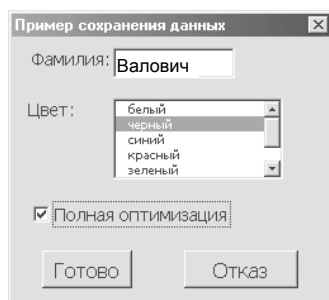


Рис. 6.21. Окно с данными перед выходом

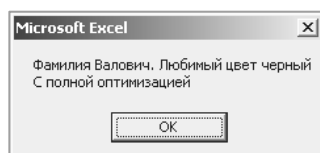


Рис. 6.22. Окно с сообщением после нажатия на кнопку *Готово*

6.2.16. Заккрытие диалогового окна

Диалоговые окна в Excel всегда работают в режиме модального диалога. Это значит, что пользователь должен полностью завершить работу в окне и закрыть его, прежде чем сможет перейти к действиям, не связанным с работой в этом окне. Для завершения работы с диалоговым окном применяется оператор *Unload* (*Выгрузить*), вызываемый при сообщении программе о желании пользователя завершить работу в этом диалоговом окне. Обычно в диалоговое окно включается одна или несколько кнопок, выбор которых приводит к завершению работы в нем (*Отказ*, *Выход*, *Готово*, *Конец работы* и т. п.).

В предыдущем примере был использован оператор *Unload* для выхода из диалогового окна после сохранения введенных пользователем данных.

Ниже приводится пример вставки данных (фамилии пользователя) из того же диалогового окна *Myform* в активный документ перед закрытием окна:

```

Private Sub cmdOK_Click()           ' нажата кнопка "Готово"
    ActiveDocument.Content.InsertAfter txtName.Text
    Unload Myform
End Sub

```

Другая возможность убрать окно с экрана – использовать метод *Hide*: *MyForm.Hide*.

6.3. Примеры макросов

6.3.1. Объектная модель Excel

Программирование на VBA можно рассматривать, как управление объектами приложения. Именно объектами и управляет это приложение. Если упростить иерархическую структуру, то это выглядит следующим образом:

```

Application
    Workbooks

```

Другими словами, главный объект – приложение (*Application*). В приложении может быть несколько книг (*Workbooks*), внутри которых находятся листы (*Worksheets* или *Sheets*). Листы разбиты на ячейки (*Cells*). При работе активными могут быть только одна книга и один лист. Рассмотрим макрос, который определяет количество листов в текущей книге:

```
Sub Test()  
    Dim book, sheet, addr, s As String  
    Dim xList As Integer  
    addr = "C"  
    book = Application.ActiveWorkbook.Name  
    sheet = Application.ActiveSheet.Name  
    Workbooks(book).Activate  
    Worksheets(sheet).Activate  
    Range("A1") = book  
    Range("B1") = sheet  
    xList = Application.Sheets.Count  
    For x = 1 To xList  
        s = addr + LTrim(Str(x))  
        Range(s) = x  
    Next x  
End Sub
```

Используя объект *Application*, получим имена текущих книг и листа. С помощью *Range* ("...") можно сослаться на ячейку. Объекты имеют огромное количество свойств и методов. Задача программиста на VBA – использовать эти свойства и методы.

При написании программы часто приходится работать с группами однотипных объектов. Для этого используются массивы или коллекции. При этом именно понятие «коллекция» в VBA получила широкое распространение. Обратиться к объекту коллекции можно, указав его номер или имя (строку) в круглых скобках после имени коллекции.

6.3.2. Коллекция *Workbooks*

В Excel на верхнем уровне иерархии объектов находится объект *Application*. Он содержит ряд коллекций. Первая коллекция – это коллекция рабочих книг *Workbooks*.

Рассмотрим пример, позволяющий узнать, сколько книг открыто:

```
Sub Test()  
    MsgBox(Str(Application.Workbooks.Count))  
End Sub
```

Функция *Str* переводит число в строку. Метод *Count* возвращает количество элементов коллекций. Метод *Open* позволяет открыть книгу. Единственный обязательный аргумент – это имя файла, остальные можно опустить:

```
Sub Test()  
    Application.Workbooks.Open("D:\Stud\My.xls")  
End Sub
```

Метод закрытия рабочей книги (т. е. удаления книги из коллекции) называется *Close*:

```
Sub Test()  
    Application.Workbooks(1).Close  
' или Application.Workbooks("My.xls").Close  
End Sub
```

Как видим, доступ можно получить по индексации и по имени книги. Следует знать, что имя книги – это имя файла, в котором он хранится. В таком варианте закроются все книги:

```
Sub Test()  
    Application.Workbooks.Close  
End Sub
```

Метод *Close* имеет необязательный аргумент типа *Boolean*. Если его установить в *True*, сделанные изменения сохраняются, в противном случае – нет. Если параметр опускается, как в примерах, приведенных выше, то при закрытии появляется диалоговое окно с вопросом о необходимости сохранения.

При работе с коллекциями часто используется цикл *For Each*. Он позволяет произвести однотипные

операции над всеми объектами:

For Each *переменная* In *коллекция*

Инструкции

Next *переменная*

Переменная должна иметь тип, совпадающий с объектами коллекции, или *Variant*, в котором можно хранить практически все. Ниже приведен пример использования типа *Variant*:

```
Sub Test()  
    Dim arrays(1) As String  
    Dim vari As Variant  
    arrays(0) = "Hello"  
    arrays(1) = "Each :-)"  
    For Each vari In arrays  
        MsgBox (vari + " - Steps")  
    Next vari  
End Sub
```

Нижеследующий пример иллюстрирует, как быстро просмотреть открытые книги:

```
Sub Test()  
    Dim vars As Variant  
    For Each vars In Workbooks  
        MsgBox (vars.Name)  
    Next vars  
End Sub
```

Или листы книги:

```
Sub Test()  
    Dim vars As Variant  
    For Each vars In Workbooks("Test.xls").Sheets  
        MsgBox (vars.Name)  
    Next vars  
End Sub
```

6.3.3. Коллекция *Sheets*

Данная коллекция представляет собой коллекцию листов (*Sheets*) в книге (*WorkBook*). Определить количество листов в книге можно с помощью следующего макроса:

```
Sub Test()  
    MsgBox (Str(Application.Workbooks("Test.xls").Sheets.Count))  
End Sub
```

Свойство *Name* позволяет получить имя листа:

```
Sub Test()  
    With Application.Workbooks("Test.xls")  
        For x = 1 To .Sheets.Count  
            MsgBox (Sheets(x).Name)  
        Next x  
    End With  
End Sub
```

Под объектом *Sheet* понимается не только рабочий лист, но и диаграмма, расположенная на отдельном листе. Так же как и лист для расчетов, диаграмма будет включена в подсчет листов. Отличить их поможет свойство *Type*:

```
Sub Test()  
    With Application.Workbooks ("Test.xls")  
        For x = 1 To .Sheets.Count  
            MsgBox (Sheets(x).Type)  
            If Sheets(x).Type = 3 Then  
                MsgBox Sheets(x).Name  
            End If  
        Next x  
    End With  
End Sub
```

Метод *Select* позволяет выделять листы:


```
Sub Test()
    Sheets("Лист2").Select
End Sub
```

К коллекции листов есть возможность добавлять свои листы. Для этого существует метод *Add*. Этот метод требует четыре аргумента: *Before*, *After*, *Count*, *Type*. Они все необязательные. Первые два отвечают за место вставки листа. Дальше количество вставляемых листов *Count* и тип листа. Типы могут быть, например, такие: *xlWorkSheet* для расчетного листа, *xlChart* для диаграммы. Если местоположение не указывать, то лист будет вставляться относительно текущего листа:

```
Sub Test()
    With Application.Workbooks.Item("Test.xls")
        Sheets.Add
    End With
End Sub
```

Метод *Parent* позволяет узнать, какой книге принадлежит текущий лист:

```
Sub Test()
    With Application.Workbooks(1)
        MsgBox (Sheets.Parent.Name)
    End With
End Sub
```

Свойство *Visible* позволяет скрыть или отобразить лист. В первом случае свойству нужно присвоить значение *True*, а во втором – *False*:

```
Sub Test()
    Sheets("Лист1").Visible = False
End Sub
```

Один из полезных методов – это метод *Copy*. Он позволяет создавать новый лист на основе существующего, т. е. использовать лист как шаблон для других листов. Аргумент *after* – это лист, после которого произойдет вставка:

```
Sub Test()
    Sheets("Лист1").Copy after:=Sheets("Лист2")
End Sub
```

У метода *Copy* есть особенность. Если не указывать аргументы, то будет создана новая книга с копируемым листом:

```
Sub Test()
    Sheets("Лист1").Copy
End Sub
```

При необходимости передвинуть лист есть метод *Move*:

```
Sub Test()
    Sheets("Лист1").Move after:=Sheets("Лист2")
End Sub
```

Метод *Delete* позволяет удалить лист:

```
Sub Test()
' Удаляет рабочий лист sSheetName,
' если дата dDelDate уже наступила
    Dim sSheetName As String
    Dim dDelDate As Date
    On Error GoTo errHandle
    sSheetName = "Лист3"
    dDelDate = #4/20/2006#
    ' Проверка даты
    If dDelDate <= Date Then
        ' Не выводить подтверждение на удаление
        Application.DisplayAlerts = False
        Sheets(sSheetName).Delete
        Application.DisplayAlerts = True
    End If
Exit Sub
```

errHandle:

```
MsgBox Err.Description, vbCritical, "Ошибка №" & Err.Number
End Sub
```

Следующая функция проверяет наличие листа с заданным именем:

```
Function IsSheetExist(ByVal shName As String) As Boolean
' Проверяет, имеется ли в активной рабочей книге лист с именем
' shName. ' В случае успеха возвращает True, иначе – False
    Dim c As Object
    On Error GoTo errHandle:
    Set c = sheets(shName)
        ' Альтернативный вариант : Sheets(shName).Select
    IsSheetExist = True
    Exit Function
errHandle:
    IsSheetExist = False
End Function
```

```
Sub Test()
    Dim sh1, sh100 As String
    sh1 = "Лист1"
    sh100 = "Лист100"
    MsgBox ("Наличие листа " & sh1 & " - " & IsSheetExist(sh1))
    MsgBox ("Наличие листа " & sh100 & " - " & IsSheetExist(sh100))
End Sub
```

6.3.4. Использование объекта Range

Объект *Range* предназначен для работы с диапазонами. Его аргументами могут быть следующие:

- строка – имя ячейки, например, Range("A1");
- строка – имя диапазона, например, Range("A1:D5");
- строковое выражение, значением которого является имя ячейки или диапазона, например, Range("A2:D" & row) или Range("A" & rowUp & ":D" & rowDown), где row, rowUp, rowDown – номера строк;
- два объекта типа *Range* – ячейки, задающие прямоугольный диапазон, например, Range(Range("A1"), Range("D5"));
- два объекта типа *Cell* – ячейки, задающие прямоугольный диапазон, например, Range(Cells(1, 1), Cells(5, 4)).

Следующий пример показывает, как изменить и получить значение ячейки:

```
Sub Test()
    Range("A1") = "Строковое значение"
    Range("A2") = 2 ' числовое значение
    Range("A2:A3") = 2 ' изменить значение всех ячеек диапазона
    Range("A4") = "=A2+A3" ' ввод формулы в ячейку
    MsgBox ("Значение в A2 = " & Range("A2"))
    MsgBox ("Значение в A2 = " & Range("A2").Value)
    MsgBox ("Значение в A4 = " & Range("A4"))
    MsgBox ("Значение в A4 = " & Range("A4").Value)
End Sub
```

Метод *Offset* позволяет сослаться на новую ячейку, относительно текущей. Первый аргумент – смещение по строке, второй – по столбцу:

```
' смещение вниз на одну ячейку и выделение ее
ActiveCell.Offset(1, 0).Select
' отрицательное число – для смещения вверх
ActiveCell.Offset(-1, 0).Select
```

Следующий пример показывает, как переместиться вниз до первой пустой ячейки:

```
Sub Test()
    Dim a As Boolean
    Dim d As Double
    Dim c As Range
    a = True
```

```

Set c = Range(ActiveCell.address)
c.Select
d = c.Value
c.Value = d
While (a = True)
    ActiveCell.Offset(1, 0).Select
    If (IsEmpty(ActiveCell.Value) = False) Then
        Set c = Range(ActiveCell.address)
        c.Select
        d = c.Value
        c.Value = d
    Else
        a = False
    End If
Wend
End Sub

```

Если надо узнать максимальные размеры листа, то для этого есть метод *UsedRange*. Он позволяет получить объект типа *Range*, из которого можно узнать максимальный столбец или строку:

```

Sub Test()
    Dim cur_range As Range
    Set cur_range = ActiveSheet.UsedRange
    Debug.Print cur_range.Address
End Sub

```

Адресовать ячейки можно и двумя цифрами по строке и столбцу. Это избавляет от утомительного разбора адресов типа \$A10. Использование *Cells(x, y)* позволяет строить удобные циклы. В следующем примере диапазон A1:E3 заполняется двузначным числом, равным индексу элемента соответствующей матрицы:

```

Sub Test()
    Dim i, j As Integer
    For i = 1 To 3
        For j = 1 To 5
            Cells(i, j) = i * 10 + j
        Next j
    Next i
End Sub

```

Границы объекта *Range* позволяет определить метод *End* с аргументом, принимающим одно из значений: *xlUp*, *xlDown*, *xlToLeft*, *xlToRight*:

```

Sub TestRange()
    Range("G2:K10") = 1
    MsgBox ("Левый верхний - " + _
        Range("I5").End(xlUp).End(xlToLeft).Address)
    MsgBox ("Правый нижний - " + _
        Range("I5").End(xlDown).End(xlToRight).Address)
    MsgBox ("Нижняя строка - " & Range("I5").End(xlDown).Row)
    MsgBox ("Правый столбец - " & Range("I5").End(xlToRight).Column)
End Sub

```

Следующий пример иллюстрирует, как выделить непустой диапазон выше текущей активной ячейки:

```

Sub SelectColumnData()
    ' что делать при ошибке
    On Error GoTo errors
    ' нижний адрес
    Dim a1 As String
    ' верхний адрес
    Dim a2 As String
    ' диапазон
    Dim ran As Range
    ' если не верхняя ячейка
    If (ActiveCell.Row <> 1) Then

```

```

' пойти вверх
ActiveCell.Offset(-1, 0).Select
' взять адрес ячейки
a1 = ActiveCell.Address
' будем подниматься
For x = 1 To (ActiveCell.Row - 1)
    ' на одну вверх
    ActiveCell.Offset(-1, 0).Select
    ' если не число выход
    If IsNumeric(ActiveCell.Value) <> True Then
        ' на одну вниз
        ActiveCell.Offset(1, 0).Select
        ' выход
        GoTo nexts
    End If
    ' если пустая
    If IsEmpty(ActiveCell.Value) = True Then
        ' на одну вниз
        ActiveCell.Offset(1, 0).Select
        ' выход
        GoTo nexts
    End If
Next x
nexts:
' получаем адрес вырехней
a2 = ActiveCell.Address
' строим диапазон
Set ran = Range(a1 + ":" + a2)
' выделяем
ran.Select
End If
' выходим из процедуры
Exit Sub
' ошибка, зовем на помощь
errors:
MsgBox "Ошибка! Сообщите разработчику."
End Sub

```

При выделении диапазона может возникнуть задача пройтись по этому диапазону с целью, например, покраски значений:

```

Sub FillColor()
    For Each c In Range("A2:D10")
        If c.Value > -5 Then
            c.Select
            With Selection.Interior
                .ColorIndex = 6
                .Pattern = xlSolid
            End With
            Selection.Font.ColorIndex = 2
        If c.Value > 5 Then
            c.Select
            Selection.Font.ColorIndex = 3
            If c.Value > 20 Then
                c.Select
                Selection.Font.ColorIndex = 4
            End If
        End If
    End If
Next c
End Sub

```

6.3.5. Обработка ошибок VBA

Для обработки ошибок в VBA есть специальный оператор *On Error*. Его задача – при возникновении

ошибки передать управление в то место процедуры, где находится обработчик ошибки. Рассмотрим следующий пример:

```
Sub Test()  
    On Error GoTo Errors1  
    Dim x, a As Integer  
    Dim c As Double  
    x = 20  
    a = 0  
    c = x / a  
    MsgBox (" Этого не должно быть")  
    Exit Sub  
Errors1:  
    MsgBox (x & " / " & a & " = Ух ты :(")  
End Sub
```

В данном примере при возникновении ошибки управление передается по метке *Errors1* и дальше выполняется код. Впрочем, прерывать процедуру из-за ошибки не всегда надо. Поэтому есть оператор *Resume Next*. Этот оператор реализует небезызвестный принцип «ни шагу назад». Выполнение пойдет дальше, несмотря на ошибку:

```
Sub Test()  
    On Error Resume Next  
    Dim x, a As Integer  
    Dim c As Double  
    x = 20  
    a = 0  
    c = x / a  
    MsgBox (" Этого не должно быть")  
    Exit Sub  
Errors1:  
    MsgBox (x & " / " & a & " = Ух ты :(")  
End Sub
```

Имеется четыре варианта оператора *Resume*: *Resume Next*, *Resume строка*, *Resume метка*, *Resume 0*. Следующий пример будет упорно требовать, чтобы ввели число, отличное от 0:

```
Sub Test()  
    Dim x, a As Integer  
    Dim c As Double  
    On Error GoTo Error1  
    x = 20  
    a = Str(InputBox("Введите число"))  
    c = x / a  
    x = 10  
    MsgBox ("Приплыли!!!")  
    GoTo Ends:  
Error1:  
    MsgBox ("Думай о программировании, а не о каникулах :)")  
    a = Str(InputBox("Введите число"))  
    Resume 0  
Ends:  
End Sub
```

6.3.6. Чтение и запись текстовых файлов

Наверно можно смело утверждать, что умение читать и записывать информацию в текстовый файл – это основа импорта и экспорта. Практически любая серьезная программа, хранящая информацию, позволяет сохранить ее в текстовом формате, какое бы он расширение не имел. Открываются файлы оператором *Open*:

```
Sub Test()  
    Open "D:\Stud\1.txt" For Input As #1  
    Close #1  
End Sub
```

Оператор *Open* может открывать файл для чтения *Input* и для записи *Output*. Цифра после ключевого слова *As* – это идентификатор файла. На его основании производится чтение и запись файла.

Следующий пример демонстрирует запись и чтение файла:

```

Sub Test()
    Open "D:\Stud\1.txt" For Output As #1
    Print #1, "Hello File"
    Close #1
    Open "D:\Stud\1.txt" For Input As #1
    Dim s As String
    Input #1, s
    MsgBox s
    Close #1
End Sub

```

Как видим, для записи можно использовать оператор *Print*, а для чтения – *Input*, воспользовавшись идентификатором открытого файла. Естественно, есть особенности работы. Например, если записать такую строку `Print #1, "Hello , File"`, то оператор `Input #1` прочитает только *Hello* и все. Запятая воспринимается как разделитель. И это правильно. Есть форматы текстовых файлов, когда числа разделены запятой. В следующем коде последовательно выведутся надписи *Hello* и *File*:

```

Input #1, s
MsgBox s
Input #1, s
MsgBox s

```

С данной проблемой можно бороться оператором *Line Input*, который прочитает строку целиком:

```

Sub Test()
    Open "D:\Stud\1.txt" For Output As #1
    Print #1, "Hello , File"
    Close #1
    Open "D:\Stud\1.txt" For Input As #1
    Dim s As String
    Line Input #1, s
    MsgBox s
    Close #1
End Sub

```

Возникает важный вопрос о том, как узнать конец файла. Для этого есть функция *EOF* (идентификатор), которая позволяет определить конец файла:

```

Sub Test()
    Dim s As String
    Open "D:\Stud\1.txt" For Output As #1
    Print #1, "Hello , File"
    Close #1
    Open "D:\Stud\1.txt" For Input As #1
    While Not EOF(1)
        Input #1, s
        MsgBox s
    Wend
    Close #1
End Sub

```

6.3.7. Работа с папками

В первом примере определим, откуда запущен Excel. У объекта *Application* есть свойство *Path*, которое и позволит нам получить информацию:

```

Sub Test()
    MsgBox (Application.Path)
End Sub

```

Операторы создания, изменения и удаления папок очень похожи на DOS-аналоги. Это `MkDir`, `ChDir` и `RmDir`. Создадим папку *MyFolder* в папке `D:\Stud`:

```

Sub Test()
    ChDir ("D:\Stud")
    MkDir ("MyFolder")
End Sub

```

И удаляем:

```
Sub Test()
  Rmdir ("MyFolder")
End Sub
```

Для получения текущей папки предназначена функция *CurDir*:

```
Sub Test()
  MsgBox (CurDir)
End Sub
```

Функция *Dir* позволяет просмотреть все файлы в папке. Только ее использование несколько специфично. Сначала *Dir* вызывается с параметрами и возвращает первое имя файла. В дальнейшем ее можно вызывать без параметров и получить следующее имя, и так до тех пор, пока не вернется пустое имя файла:

```
Sub Test()
  Dim s As String
  s = Dir("D:\Stud"\"*.*)
  Debug.Print s
  Do While s <> ""
    s = Dir
    Debug.Print s
  Loop
End Sub
```

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. **Астапкина, Т. В.** Windows 95, Word 7.0, Excel 7.0 : учеб. пособие / Т. В. Астапкина, Л. М. Ашарчук, И. В. Дубинина. – Гомель : ГКИ, 1999.
2. **Буза, М. К.** Операционная среда Windows и ее приложения / М. К. Буза, Л. В. Певзнер, И. А. Хижняк. – Минск : Выш. шк., 1997.
3. **Джонс, Э.** Библия пользователя Microsoft Office Professional для Windows 95 / Э. Джонс, Д. Саттон. – Киев : Диалектика, 1996.
4. **Левчук, Е. А.** Построение информационно-логистической модели данных и ее реализация в СУБД Access : пособие / Е. А. Левчук, Т. А. Заяц. – Гомель : Бел. торгово-экон. ун-т потребит. кооп., 2003.
5. **Меллер, А.** Office 97 : [пер. с нем.] / А. Меллер, К. Пейтон. – М. : БИНОМ, 1999.
6. **Основы информатики и вычислительной техники : программа курса** / С. М. Мовшович [и др.]. – Гомель : Бел. торгово-экон. ун-т потребит. кооп., 2001.
7. **Основы экономической информатики** / под ред. А. Н. Морозевича. – Минск : БГЭУ, 1998.
8. **Экономическая информатика : учеб. для вузов** / под ред. В. В. Евдокимова. – СПб. : Питер Паблишинг, 1997.

СОДЕРЖАНИЕ

Введение	4
Раздел 1. Использование инструментария Excel для решения экономических задач	4
1.1. Назначение и классификация электронных таблиц	4
1.2. Критерии качества разработки электронной таблицы	4
1.3. Понятие функции	5
1.4. Ввод функции	6
1.5. Встроенные функции	7
1.6. Типы ошибок в формулах Excel	14
1.7. Связывание и консолидация рабочих таблиц	14
1.8. Создание и использование формул массивов	18
1.9. Форматы данных, поддерживаемые Excel	20
1.10. Импортирование данных	21
1.11. Работа со списками	25
1.12. Анализ данных с помощью сводных таблиц	30
1.13. Анализ данных с использованием надстроек Excel	35
1.14. Базовые сведения о VBA	39
1.15. Интегрированная среда разработки	41
1.16. Отладка программ	44
1.17. Основы программирования на VBA	47
Раздел 2. Использование функций Excel в экономических расчетах	52
2.1. Совместное использование логических функций	52
2.2. Учет доходов за обучение	52
2.3. Использование функции СУММЕСЛИ	55
2.4. Использование смешанных ссылок	56
2.5. Расчет итогов с условием	57
2.6. Учет расходов и налогов с накоплением	58
2.7. Использование формул массивов	62
2.8. Использование консолидации данных	65
Раздел 3. Планирование на малых и средних предприятиях средствами Excel	67
3.1. Постановка задачи	67
3.2. Предплановые расчеты	69
3.3. Планирование наличия мощности	70
3.4. Планирование потребности в мощности	73
3.5. Планирование потребности в материалах	76
3.6. Планирование численности персонала	77
3.7. Планирование фонда оплаты труда	79
3.8. Планирование себестоимости товарной продукции	81
3.9. Планирование прибыли и рентабельности	84
Раздел 4. Импортирование данных в MS Excel из внешних источников	85
4.1. Основные теоретические сведения	85
4.2. Использование надстройки MS Query для импортирования в MS Excel данных, созданных в СУБД Access	85
4.3. Задания для самостоятельной работы	99
Раздел 5. Использование надстройки MS Excel «Поиск решения» в экономических расчетах	99
5.1. Основные теоретические сведения	99
5.2. Применение средства поиска решения для определения оптимального количества выпускаемых изделий с максимальной прибылью	100
5.3. Задачи для самостоятельного решения	105
Раздел 6. Введение в VBA для Excel	115
6.1. Интегрированная среда разработки VBA для Excel	115
6.2. Создание диалоговых окон на VBA	124
6.3. Примеры макросов	135
Список рекомендуемой литературы	144

Учебное издание

Левчук Елена Аркадьевна
Заяц Татьяна Александровна

АВТОМАТИЗАЦИЯ ЭКОНОМИЧЕСКИХ РАСЧЕТОВ

Пособие
для студентов экономических специальностей

Редактор Н. В. Славницкая
Технический редактор И. А. Козлова
Компьютерная верстка Л. Ф. Кириленкова

Подписано в печать 05.03.08. Бумага типографская № 1.
Формат 60 × 84 ¹/₁₆. Гарнитура Таймс. Ризография.
Усл. печ. л. 14,65. Уч.-изд. л. 15,74. Тираж 150 экз.
Заказ №

Учреждение образования «Белорусский торгово-экономический
университет потребительской кооперации».
246029, г. Гомель, просп. Октября, 50.
ЛИ № 02330/0056814 от 02.03.2004 г.

Отпечатано в учреждении образования «Белорусский торгово-экономический
университет потребительской кооперации».
246029, г. Гомель, просп. Октября, 50.